Model 472
Multibus to Pertec-formatted
Tape Controller User's Manual

Your Partner for Performance

166-018-001
Revision G
March 27, 1990

472 Revision Level History

Revision	Description
A (9/1/83)	Initial release.
B (2/1/84)	General rewrite of Rev. A Manual.
C (1/85)	Technical rewrite; consolidated manual; deleted the subfunction codes that Xylogics does not support.
C1 (7/22/85)	Technical additions: expanded Sections 2.6.3 and 2.5.8.3.
Rev. D (5/16/86)	Incorporated Rev. G microcode: affected sections include 2.4.2 (Table 2-4), 2.5.2.5, 2.5.3.4, 2.5.5, 2.5.6.2, 2.5.8, and 2.5.8.2. Revised subsection numbers in Section 2.5. Added Section 3.8.
Rev. D1 (12/22/87)	Reorganized manual. Section 2 expanded into more managable sections: Section 2.4 now Section 3; Section 2.5 now Section 4; Section 2.6 now Section 5; Section 3 now Section 6; and Section 4 now Section 7.
Rev. E (3/14/88)	Revised Table 3-4 (Summary of Completion Codes: Codes 07 and 24-2F) and related completion code descriptions.
Rev. F (4/24/88)	Revised Interrupt Mode (IOPB Byte 05) and related bit descriptions in Section 3. Revised all IOPB diagrams (except Command Code B) in Sections 3 and 4 to incorporate this change. Revised style of charts and tables.
Rev. F1 (10/20/88)	Clarified Section 6.2.2 (20/24-Bit Address Relocation) and corrected Section 6.2.3 (24-Bit Extended Addressing).
Rev. F2 (3/24/89)	Corrected Table 4-4 (Tape Command Lines: Select 25/100 IPS). Section 7.2: Pertec cables J1/J2.
Rev. G (3/27/90)	Removed Section 5.4 (Error Recovery); incorporated the information into Section 3.1.4.1. (Completion Code Descriptions). Added error code 34 to Section 3.1.4.1. Clarified Section 6.2.1 (Base Address Selection). Added Figure 6-2 (Base Address Jumpers); replaced Table 6-1 (Base Address Selection) with Figure 6-3 (Jumpering the Base Address).

Table of Contents

Table of Contents

Table of Contents

Table of Contents

List of Figures and Tables

1

Specifications

1

Specifications

1.0 General

The Xylogics Model 472 tape controller connects up to eight Pertec-formatted ½-inch streaming or start/stop tape drives to IEE P796 Multibus® systems. The 472 implements data transfers via Direct Memory Access (DMA), allowing maximum throughput; it implements system control via Input/Output Parameter Blocks (IOPBs) and byte Input/Output Registers. The 472 circuitry includes a DMA sequencer and a microprocessor for control. The 472 is the fastest tape controller available for Multibus systems with a DMA transfer rate capability of 3.0 megabytes per second (MBS).

1.1 Using This Manual

This manual provides two system software reference cards that display the IOPB structure and codes (see inserts). Section 2 is a programming reference, Section 3 describes the IOPB structure, and Section 4 describes the 472's commands. Section 5 gives a good overview of the 472's programming procedures. Section 6 details how to install and test the 472 controller, and Section 7 includes drive interface information.

1.1.1 Abbreviations

This manual uses the following mnemonics:

AACK	Attention Acknowledge
ADRM	Addressing Mode
AREQ	Attention Request
AUD	Auto-update
BOT	Beginning of Tape
BPI	Bits Per Inch
BWM	Byte/Word Mode
CER	Corrected Error
CHEN	Chain Enable
CONS	Consecutive
CPU	Central Processing Unit and/or Computer
CSR	Control and Status Register

Multibus is a registered trademark of Intel Corporation.

1.1.1 Abbreviations (continued)

DBY	Data Busy
DERR	Double Error
DLD	Data Late Detected
DMA	Direct Memory Access
ENT	End of Tape
ERR	Error
ERRS	Error Summary
FBY	Formatter Busy
FIFO	First In/First Out Buffer
FMK	File Mark Detected
FPT	Write-protected
GBSY	Go/Busy
GCR	Group Coded Recording
H	Notation for hexadecimal values
HER	Hard Error
IBG	Interblock Gap
IEI	Interrupt on each IOPB
IEN	Interrupt Enable
I/O	Input/Output
IOPB	Input/Output Parameter Block
IPND	Interrupt Pending
IPS	Inches Per Second
KB	Kilobyte
KBS	Kilobytes Per Second
MB	Megabyte
MBS	Megabytes Per Second
NRZI	Non-return to Zero
ONL	On-line/Off-line
PE	Phase Encoded
RAM	Random Access Memory
RDY	Drive Ready
RELO	Relocate
RETY	Retry
REW	Rewind
REV	Reverse Operation
RLL	Record Length Long
RLS	Record Length Short
SWAP	Swap Byte

1.2 Design Reliability

The following Xylogics features minimize the likelihood of product failure:

- Low parts count through microprogramming
- Low power Schottky integrated circuits
- Low stress design on all components
- All components burned-in
- One card in backplane or expansion chassis
- Power-cycling under thermal stress during test

1.3 Physical

Packaging: The 472 completely resides on one printed circuit board. It plugs into any 16-, 20-, or 24-bit Intel Multibus or IEEE P796 card cage.

Dimensions: The 472 measures 12-inches long by 6.75-inches high (30.48 cm by 17.15 cm); it is identical in form-factor to the standard Intel Multibus and IEEE P796 printed circuit board.

Shipping Weight: 8 pounds (3.6 kg).

1.4 Environmental

The 472 environmental requirements are similar to the Intel 86/12 SBC or equivalent Multibus processors (typically 0–55°C; up to 90% relative humidity without condensation).

1.5 Electrical

Power: The 472 requires 5.0 amperes at +5 VDC.

Tolerance: Voltages *must* be within ± 5% (4.75 to 5.25).

Grounding: Common earth ground must be established between the tape drives and the CPU chassis, backplane, and expansion cabinets.

1.6 System-related Specifications

Transfer Control: Direct Memory Access (DMA).

DMA Throttle Control: Programmable throttle supports any Multibus throughput speed.

Interrupt Priority: INT5/ standard; others jumper controlled.

Interrupts: Non-bus vectored.

Control Technique: Channel driven control; programmable microprocessor.

Addressing Capability: 16-, 20-, and 24-bit.

Controller IOPB Length: 18 bytes.

Controller Registers: Six 8-bit I/O Registers.

I/O Addressing Capability: The 472 decodes byte addresses for its on-board registers; it responds to 8- or 16-bit I/O addresses.

Data Transfer Modes: The 472 transfers data in 8-bit bytes or 16-bit words.

Data Buffering: On-board FIFO memory accommodates 8K bytes in word mode.

Data Transfer Limit: From 1 to 65,535 bytes.

Software Support: Standard software drivers supplied for use in UNIX® or RMX-86® systems, and UNIX-like systems (source included).

Diagnostic Support: Comprehensive set of stand-alone diagnostics written in the *C* programming language are available.

Test Station: The Xylogics Customer Acceptance Tester (XYCAT) supports the 472 with stand-alone diagnostics for incoming inspection and field testing. This is a single Multibus board with CPU, RAM- and ROM-based diagnostics written in *C*.

Status LEDs: The 472 implements two status LEDs. L1 indicates successful completion of on-board diagnostics; L2 indicates the controller is active.

DMA Data Transfer Rate: The 472 transfers data at a rate of up to 3.0 MBS to the system bus, assuming a memory XACK response time of 300 ns, however the actual data rate depends on the individual CPU memory response time. With appropriate memory, the 472 runs a streaming 6250 BPI GCR tape at 100 IPS.

1.7 Tape Drive-related Specifications

Tape Interface: Industry standard Pertec-formatted interface.

Maximum Tape Capacity: More than 2.4 gigabytes of on-line storage (drive limitation).

Number of Tape Drives: The 472 supports up to eight tape drives, including any mix of capacities or speeds.

Tape Speed: The 472 supports tape speeds ranging from 12.5 IPS to 200 IPS (software programmable).

Tape Density: The 472 supports tape densities of 800 BPI, 1600 BPI, 3200 BPI and 6250 BPI (software programmable).

Data Verification: 800 BPI horizontal and vertical parity (NRZI). 1600/3200 BPI Phase Encoding (PE). 6250 BPI Group Encoding (GCR).

Recording Mode: 9-track; ANSI- and IBM-compatible.

Tape Data Transfer Rate: Depends on particular tape drive speed and density; 6250 BPI at 100 IPS yields a 625 KBS data rate.

Cabling: Standard Pertec-formatted interface comprising two 50-pin flat ribbon cables.

1.8 Programmable Features

- 16- or 20/24-bit address bus support
- 8- or 16-bit data transfers
- Interrupt or polled operation
- DMA throttle
- Tape drive speed and density

1.9 Internal Registers

Section 3 describes specific bits within the I/O Registers. The software driver establishes commands by loading and reading these registers (see Table 2-1).

1.10 Command Technique

The 472 command technique allows IOPB chaining with concurrent host and controller operations. Channel control allows a software driver to establish a tape command and parameters in an IOPB in system memory. The software driver initiates a single IOPB, or an IOPB chain, by loading the memory address of the first (chained) IOPB into the 472 Relocation and Address Registers and setting the *Go/Busy* (GBSY) bit in the Control and Status Register (CSR). GBSY remains set until the 472 completes the command(s) or an error occurs.

The 472 reads the IOPB from system memory via Direct Memory Access (DMA) and performs the required function. When the 472 completes an IOPB, or detects an error, it writes the status and a completion code into Bytes 3 and 4 of the related IOPB. Setting the *Error Status* (ERRS) bit in the CSR usually clears hard errors.

1.10.1 Command-Chaining

The 472 provides an inherent IOPB chaining capability for complex operations. The software driver sets up a string of commands (e.g., disk-to-tape copy) that execute a series of tape operations without operating system intervention. At any time, system software can add new IOPBs and/or remove completed IOPBs from the chain via the attention protocol. You may implement overlap rewind operations in multidrive systems by setting the *Chain Enable* (CHEN) and *Enable Extended Function* (EEF) bits.

2

Programming Reference

2

Programming Reference

2.0 General

This section describes programming procedures for the 472 streaming tape controller. The 472 easily connects many different processors to a wide variety of tape drives.

2.1 Programming Techniques

Set up the 472 commands by preparing an IOPB in system memory. Initiate a command by loading the IOPB address into the 472's registers and setting GBSY in the CSR. GBSY remains set until the controller completes all the commands in the IOPB chain, or a hard error occurs. When the 472 completes a command, it writes the corresponding status and completion codes into Bytes 2 and 3 of the completed IOPB in system memory. Table 3-1 lists the IOPB bytes.

The IOPB, located in system memory, contains the 472 command parameters. The CPU writes and reads the IOPB with normal byte or word instructions. The 472 reads and writes the IOPB with byte mode Direct Memory Access (DMA).

System software builds an IOPB in host memory with the appropriate information and then passes the IOPB address by loading the I/O Registers. The software driver then sets GBSY. The 472 transfers the IOPB from memory, processes the command, and clears GBSY. While processing the command, the 472 may access the IOPB again and it may also DMA data to or from memory. Software may chain IOPBs together. IOPB chaining allows the 472 to transfer data without operating system intervention.

Each byte in the IOPB has an address relative to the Command byte. Reserving all 18 bytes of allowable IOPB space maintains IOPB integrity.

2.2 Multibus Address Relocation

The 472 accesses Multibus memory via address relocation. Address relocation involves adding two addresses to form a larger physical address. The 472 supports two types of address relocation: 20- and 24-bit relocation. Use either type of relocation when specifying 16-bits of memory address. Load the Relocation Register with zero for 16-bit memory addressing. A jumper on the 472 board selects either 20- or 24-bit relocation. Bit 3 (ADRM) in the CSR indicates the addressing mode. When set, the board is jumpered for 24-bit relocation.

> *This manual refers to both IOPB relocation and data relocation. Do not confuse them. IOPB relocation refers to the address at which the IOPB resides in memory. Data relocation refers to the address at which the data buffer exists. Data relocation may be affected by bit 6 (RELO) of Command Byte 0, but IOPB relocation is not. The jumper for 20/24-bit address selection affects address relocation for both data and IOPBs.*

2.2.1 20-Bit Address Relocation

The 472 forms a 20-bit physical address by adding a 16-bit address word to a shifted 16-bit relocation word. The relocation word shifts by four bits.

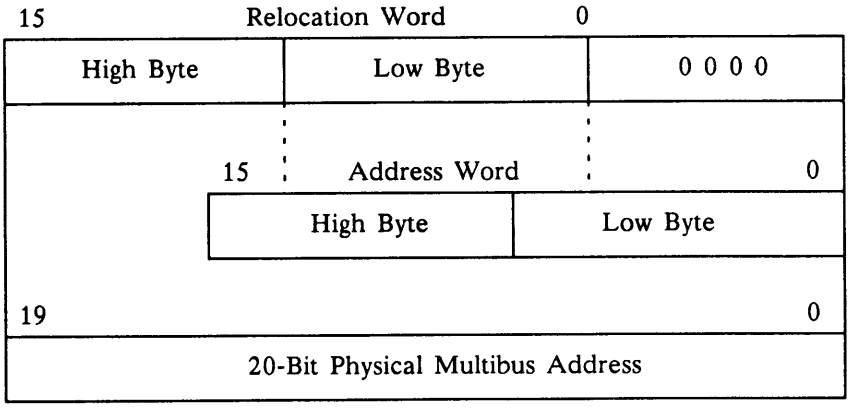


Figure 2-1. 20-Bit Multibus Address Relocation

2.2.2 24-Bit Address Relocation

The 472 calculates a 32-bit physical address for 24-bit address relocation. The address word comprises the least significant 16 bits; the relocation word comprises the most significant 16 bits. When addressing memory, the 472 only uses the lower 24 bits of the physical address.

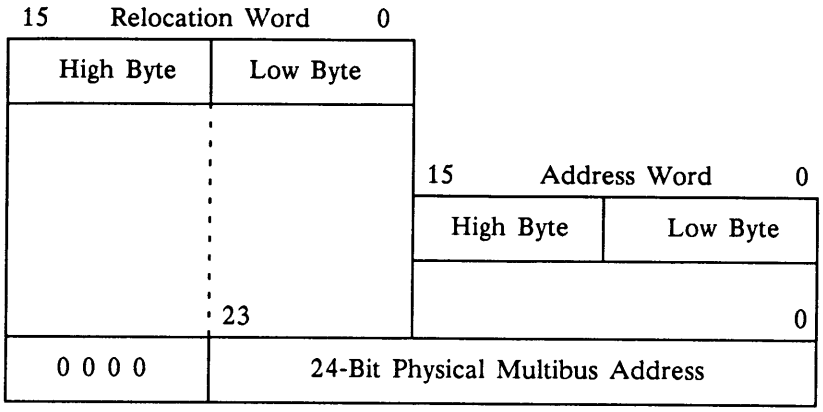


Figure 2–2. 24-Bit Multibus Address Relocation

2.2.3 IOPB Address Relocation

IOPB relocation occurs whenever a non-zero value is loaded into the IOPB Relocation Registers. The 472 combines the IOPB Address Registers and IOPB Relocation Registers to form a 20- or 24-bit physical memory address.

When chaining IOPBs, the 472 uses the IOPB Relocation Registers and the Next IOPB Address bytes to from a new 20- or 24-bit physical Multibus address. This address points to the next IOPB in the chain. All IOPBs in a chain must reside in the same 64K-byte segment whose base address is in the Relocation Registers. The 472 computes the base address by shifting the Relocation Registers 4 or 16 bits to the left, depending on the relocation mode (see Figure 2–1 and Figure 2–2).

2.2.4 Data Transfer Address Relocation

IOPB Bytes A through D specify the starting memory address for a data transfer operation. If RELO is clear, the Data Address bytes (IOPB Bytes A and B) specify the transfer's physical Multibus address. If RELO is set, the 472 uses Bytes C and D as the Data Relocation bytes, and Bytes A and B as the Data Address bytes. Data relocation occurs in the same manner as IOPB relocation.

2.3 Input/Output Registers

Table 2-1. Input/Output Register Addresses (Hex)

Use	8-Bit	16-Bit
IOPB Relocation Register Low Byte	60	EE60
IOPB Relocation Register High Byte	61	EE61
IOPB Address Register Low Byte	62	EE62
IOPB Address Register High Byte	63	EE63
Control and Status Register (CSR)	64	EE64
Controller Reset/Update Register	65	EE65

2.3.1 Input/Output Register Addressing

The 472 Input/Output Registers are addressed as input/output byte ports on the Multibus. The I/O Registers use a standard base address of 60H or EE60H. Table 2-1 summarizes the 472 I/O Registers (see Section 6.2.1 for alternate base addresses).

2.3.2 Relocation Registers

There are two Relocation Registers: one contains the low byte of the relocation address, the other contains the high byte. The two Relocation Registers are the most significant portion of the IOPB memory address. The 472 clears these registers on power-up. Set them to zero when using 16-bit addresses. Writing anything but zero to these registers causes IOPB relocation (see Figure 2-1 and Figure 2-2).

2.3.3 Address Registers

There are two Address Registers: one contains the low byte of the IOPB address, the other contains the high byte. These registers are the least significant portion of the IOPB memory address. The 472 clears these registers on power-up.

2.3.4 Controller Reset/Update Register (I/O Address 45 or EE45)

The 472 executes a *Controller Reset* when reading the Controller Reset/Update Register (i.e., the 472 clears the registers along with IPND, ERR, and DERR; selects Drive 0, sets GBSY, and latches the *ready* status). Wait for GBSY to clear before issuing commands.

When the 472 writes the Controller Reset/Update Register (actual data written is insignificant), it updates the IOPB whose address is currently stored in the Address and Relocation Registers. The Update IOPB command writes the information from the 472's internal registers to the current IOPB. GBSY remains set until the 472 completes the update.

2.3.5 Control and Status Register

7	6	5	4	3	2	1	0
Go/Busy	General Error	Double Error	Interrupt Pending	Addressing Mode	Att. Request	Att. Acknowl-edge	Drive Ready

Bit	Mnemonic	Description
7	GBSY	**Go/Busy:** Setting GBSY starts a transfer. When set, the 472 is busy executing a command. GBSY remains set until the 472 completes the current IOPB (chain). When clear, the 472 is ready for another IOPB. (Read and write access.)

2.3.5 Control and Status Register (continued)

Bit	Mnemonic	Description
6	ERR	**General Error:** Sets when the 472 encounters a hard error. Clear ERR before executing another command (write a one to ERR [*Error Reset*] or execute a *Controller Reset*). (Read and write access.)
5	DERR	**Double Error:** When set, an error occurred and a previous error condition has not been cleared. This usually means that the 472 cannot properly DMA the status bytes to memory. Executing an *Error Reset* or a *Controller Reset* clears a single or double error. If DERR is set, but ERR is clear, the status bits in the CSR are invalid. The program interprets CSR status *after* DERR clears. (Read access.)

> *It is more efficient to clear an error on the 472 by executing an Error Reset (writing a one to ERR) than by executing a Controller Reset.*

4	IPND	**Interrupt Pending:** Sets when the 472 completes an IOPB, interrupts, and the interrupt is not serviced. Clear this condition before executing another command (except IOPB update) by executing an Interrupt Reset (write a one to IPND) or by executing a *Controller Reset*. System software may only write IPND and AREQ in the CSR while the 472 is busy. (Read and write access.)

> *It is more efficient to acknowledge an interrupt by executing an Interrupt Reset than by executing a Controller Reset.*

3	ADRM	**Addressing Mode:** When set, the 472 is in 24-bit addressing mode. When clear, the 472 is in 20-bit addressing mode. A hardware jumper on the 472 board selects the addressing mode; it is not software programmable (see Section 3.2.2). (Read access.)

2.3.5 Control and Status Register (continued)

Bit	Mnemonic	Description
2	AREQ	**Attention Request:** System software sets AREQ (to gain the attention of the 472 when it is busy processing commands) and waits until the 472 acknowledges the request with AACK. After the 472 sets AACK, system software may remove completed IOPBs and/or add new IOPBs. When system software completes work on the IOPB chain, it clears AREQ and the 472 clears AACK and resumes operation. (Read and write access.)
1	AACK	**Attention Acknowledge:** The 472 sets AACK to acknowledge an AREQ by system software. The 472 clears AACK after system software clears AREQ. If *Interrupt on each IOPB* (IEI) is set, an interrupt occurs when the 472 sets AACK. (Read access.)
0	DRDY	**Drive Ready:** Sets when the currently selected drive is on-line. After a *Controller Reset*, the 472 automatically selects Drive 0 and latches its *ready* status. Executing any IOPB selects the drive specified in the *Unit Select* byte and displays the selected drive status in DRDY. (Read access.)

> *While the 472 is busy, only IPND and AREQ have write access to the registers. Any other access attempts result in a busy conflict error.*

2.3.6 Control and Status Register Read Protocol

The CSR includes a Write Register (Control) and a Read Register (Status). When a Write operation accesses the CSR, the 472 posts the status code A0H (invalid status), indicating the 472 is busy updating its internal registers. After the 472 updates its internal registers, it clears A0H from the CSR and posts a valid status code. As long as the CSR status remains invalid (A0H), any attempt to write the CSR returns an error. Wait until the 472 posts a valid status code in the CSR before performing a bit test. A CSR read in an interrupt handler must follow this protocol even if not preceded by a CSR write (the CSR may have been written prior to the interrupt; see Figure 2-3).

2.3.6 Control and Status Register Read Protocol (continued)

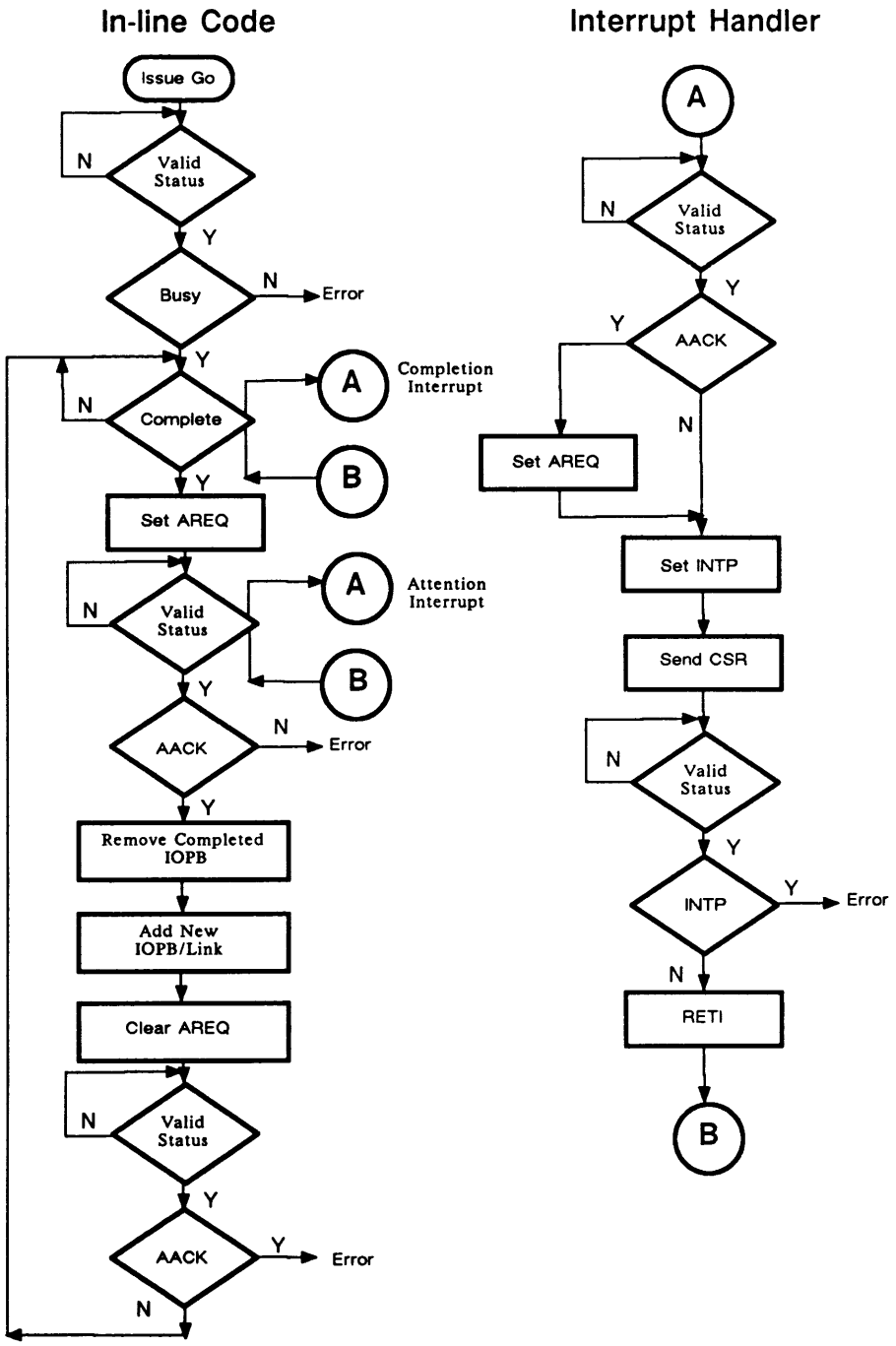


Figure 2-3. Typical IOPB Chain Handler

3

IOPB Description

3

IOPB Description

3.0 General

The Input/Output Parameter Block (IOPB) passes messages between the 472 and host software. Software passes the type of transfer, the data addresses, and the count to the 472. The 472 returns the transfer status and, if AUD is set or an error occurs, the ending addresses upon command completion. Each IOPB is 18 bytes long.

> *All IOPBs in a chain must exist in the same 64K-byte segment whose base address is specified in the Relocation Registers.*

Table 3-1. IOPB Byte Definitions

Byte Address (Hex)	Description	Mnemonic
0	Command	COMM
1	Subfunction	SUBFUN
2	Status Byte 1	STAT1
3	Status Byte 2	STAT2
4	Status Byte 3	STAT3
5	Interrupt Mode	IMODE
6	Throttle	THROT
7	Unit Select	DRIVE
8	Count Low	CNTL
9	Count High	CNTH
A	Data Address Low	DATAL
B	Data Address High	DATAH
C	Data Relocation Low	DATARL
D	Data Relocation High	DATARH
E	Next IOPB Address Low	NIOPL*
F	Next IOPB Address High	NIOPH*
10	Actual Count Low	ACNTL
11	Actual Count High	ACNTH

3.1 Standard IOPB

	7	6	5	4	3	2	1	0	
00	AUD	RELO	CHEN	IEN	Command Code				COMM
01	SWAP	RETY	REV	CONS	Subfunction Code				SUBFUN
02	ERRS	0		Controller Type			0	Done	STAT1
03	Error or Completion Code								STAT2
04	EOT	BOT	FPT	REW	ONL	RDY	DBY	FBY	STAT3
05	HER	IEI	GCR	FMK	DLD	RLL	RLS	CER	IMODE
06	BWM	0				Throttle			THROT
07	0					Unit Select			DRIVE
08	Count Low								CNTL
09	Count High								CNTH
0A	Data Transfer Address Low								DATAL
0B	Data Transfer Address High								DATAH
0C	Data Transfer Relocation Address Low								DATARL
0D	Data Transfer Relocation Address High								DATARH
0E	Next IOPB Address Low								NIOPL*
0F	Next IOPB Address High								NIOPH*
10	Actual Count Low								ACNTL
11	Actual Count High								ACNTH

3.1.1 Command Byte (IOPB Byte 0)

7	6	5	4	3	2	1	0
Auto-update	Data Reloc.	Chain Enable	Interrupt Enable	Command			

Bit	Mnemonic	Description
7	AUD	**Auto-update:** When clear, the 472 only updates Status Bytes 1 and 2. When set, the 472 updates the current IOPB upon its completion. All commands update Status Bytes 1, 2, and 3. Read or Write commands update the data and relocation addresses and point to the end or beginning of the memory buffer depending on the direction of the command. The 472 updates the Actual Count bytes if the issued command supports them.
6	RELO	**Relocation:** When clear, the 472 generates Multibus data addresses as 16-bit values, sets bits 16 through 23 to zero, and ignores the Data Relocation Address bytes. If set, the 472 enables data relocation, and forms physical Multibus addresses (see Figure 2-1 and Figure 2-2). IOPB relocation occurs whenever the IOPB Relocation Registers are non-zero.
5	CHEN	**Chain Enable:** When clear, the 472 executes the current IOPB and clears GBSY upon completion. When set, the 472 processes the next IOPB. The Next IOPB Address bytes and the Relocation Registers specify the new IOPB address.
4	IEN	**Interrupt Enable:** When clear, the 472 does not generate interrupts. When set, the 472 generates appropriate interrupts as defined by the Interrupt Mode byte.
3-0	COM	**Command:** Table 3-2 lists the 472 commands.

3.1.1 Command Byte (IOPB Byte 0) (continued)

Table 3-2. Controller Commands

Value (Hex)	Bits: 3 2 1 0	Command
0	0 0 0 0	No Operation
1	0 0 0 1	Write
2	0 0 1 0	Read
3	0 0 1 1	Reserved
4	0 1 0 0	Reserved
5	0 1 0 1	Position
6	0 1 1 0	Drive Reset
7	0 1 1 1	Write Tape Mark/Erase
8	1 0 0 0	Reserved
9	1 0 0 1	Get Status
A	1 0 1 0	Reserved
B	1 0 1 1	Set Parameters
C	1 1 0 0	Self Test
D	1 1 0 1	Reserved
E	1 1 1 0	Reserved
F	1 1 1 1	Reserved

3.1.2 Subfunction Code (IOPB Byte 1)

Each command may have one or more associated subfunctions. This byte has both bit encoded and bit flag fields associated with the actual subfunction code.

7	6	5	4	3	2	1	0
Swap Byte	Retry	Reverse	Con-secutive	Subfunction Code			

3.1.2 Subfunction Code (IOPB Byte 1) (continued)

Bit	Mnemonic	Description
7	SWAP	**Swap Byte:** When set, the 472 performs the operation specified in the subfunction code but swaps the most significant byte in the word with the least significant byte. This feature enables data transfer compatibility with machines that order their bytes in a word opposite that of the Multibus. When clear, the 472 orders the bytes as they normally appear in Multibus memory (see Section 3.5).
6	RETY	**Retry:** When set, the 472 retries the requested operation up to four times before returning a hard error code.
5	REV	**Reverse:** When set, the 472 performs the operation in the reverse direction on the tape.
4	CONS	**Consecutive:** When set, the 472 searches for consecutive tape marks; the count field must issue the required count. The Position command uses CONS to easily locate the logical end of the tape.
3	SUBF	**Subfunction Code:** Most commands have associated subfunction codes. The command defines the actual subfunction code.

Table 3-3. Subfunction Codes

Command Code	Command	Subfun Code	Description
0	NOP	00	No operation
1	Write	00	Write
		01	Write Edit
		40	Write Retry
		80	Write Swap Byte
		C0	Write Retry Swap Byte
2	Read	00	Read Next
		20	Read Previous
		21	Read Reverse Edit
		40	Read Next Retry
		60	Read Previous Retry
		80	Read Next Swap Bytes
		A0	Read Previous Swap Bytes
		C0	Read Next Retry Swap Bytes
		E0	Read Prev. Retry Swap Bytes

3.1.2 Subfunction Code (IOPB Byte 1) (continued)

Table 3-3. Subfunction Codes (continued)

Command Code	Command	Subfun. Code	Description
5	Position	00	Space Records Forward
		20	Space Records Reverse
		01	Skip Tape Marks Forward
		02	Rewind
		03	Unload
		11	Search Consecutive Forward
		31	Search Consecutive Reverse
6	Drive Reset	00	Drive Reset
		01	Load On-line
7	Write Tape Mark/	00	Write Tape Mark
	Erase	01	Erase
		02	Security Erase
		03	Variable Erase
		40	Write Tape Mark Retry
9	Get Status	00	Completion Codes
		01	Read Sense Bytes
		02	Read Extended Sense Bytes
B	Set Parameters	00	Set Low Density Mode
		01	Set High Density Mode
		02	Set Low Speed
		03	Set High Speed
		04	Long Gap Mode Select
		05	Long Gap Mode Disable
		06	Set 800 BPI
		07	Set 3200 BPI
		08	Set 6250 BPI
		09	Set 25 IPS
		0A	Set 100 IPS
		0B	Megatape Track Select
		0F	Command Pass Through

If retries are enabled, the 472 automatically attempts the operation up to four times, unless the error clears. If the error does not clear, the 472 terminates IOPB processing (general error). If retries are disabled, all hard errors terminate IOPB processing.

3.1.3 Status Byte 1 (IOPB Byte 2)

7	6	5	4	3	2	1	0
Error Summary	Reserved		Controller Type			Reserved	Done

Bit	Mnemonic	Description
7	ERRS	**Error Summary:** Sets when the 472 encounters a hard error during IOPB processing; does not set on a soft error. Clear indicates successful IOPB completion.
6-5		**Reserved.**
4-2	CTYP	**Controller Type:** Xylogics assigns each Multibus controller a controller type code as follows:

Bit 4	Bit 3	Bit 2	Controller
0	0	0	440
0	0	1	450
0	1	0	472

1		**Reserved.**
0	DONE	**Done:** The 472 sets *Done* when the IOPB is complete. Status Byte 2 holds the IOPB's completion code.

Software must clear Status Bytes 1 through 3 before passing the IOPB to the 472.

3.1.4 Status Byte 2 (IOPB Byte 3)

After the 472 executes the IOPB, Status Byte 2 contains its completion code. Table 3-4 lists the completion codes. Table 3-5 summarizes the 472 error reporting hierarchy. The following sections describe completion codes, along with any required corrective action. Unless otherwise noted, executing either an *Error Reset* or a *Controller Reset* clears a hard error. Hard errors terminate IOPB processing. To continue IOPB processing, clear the hard error and restart the controller. Status errors do not terminate IOPB processing.

Table 3-4. Summary of Completion Codes

Code	Type	Definition
00	Status	Successful Completion - No Errors
01	Hard	Interrupt Pending
02	-	Reserved
03	Hard	Busy Conflict
04	Hard	Operation Timeout
05	-	Reserved
06	Hard	Hard Error
07	Hard	Interface Parity Error
08-0D	-	Reserved
0E	Hard	Slave ACK Error (Non-existent memory)
0F-13	-	Reserved
14	Hard	Write-protect Error
15	Hard	Unimplemented Command
16	Hard	Drive Off-line
17-19	-	Reserved
1A	Hard	Self Test A Failed
1B	Hard	Self Test B Failed
1C	Hard	Self Test C Failed
1D	Hard	Tape Mark Failure
1E	Hard	Tape Mark Detected on Read
1F	Status	Corrected Data
20-21	-	Reserved
22	Hard	Record Length Short
23	Hard	Record Length Long
24-2F	-	Reserved
30	Hard	Reverse Into BOT
31	Hard	EOT Detected
32	Status	ID Burst Detected
33	Hard	Data Late Detected
34	Hard	Bad ID Field Detected

3.1.4 Status Byte 2 (IOPB Byte 3) (continued)

Table 3-5. 472 Error Reporting Hierarchy

Priority	Order	Error
Lowest	First Check	ID Burst
	Second Check	Corrected Data*
	:	EOT
	:	Hard Error
	:	Record Length Long*
	:	Record Length Short*
	:	Reverse Into BOT*
	:	Data Late Detected
	:	Tape Mark Detected*
	:	Operation Timeout
Highest	Last Check	Slave Acknowledge

* *Checked only during a Read command*

3.1.4.1 Completion Code Descriptions

Code(H) Description

00 *Successful Completion:* Not an error; the 472 successfully completed the command; software may remove the IOPB from the queue.

01 *Interrupt Pending Error:* The 472 attempted an operation with a previous interrupt still pending. Only Interrupt Acknowledge, Update IOPB, Controller Reset, or Error Reset operations are allowed while an interrupt is pending. Do not retry the operation.

02 *Reserved.*

03 *Busy Conflict:* A register write is attempted while GBSY is set. Only AREQ and IPND have write access while the 472 is busy. Do not retry the operation.

04 *Operation Timeout:* The 472 did not complete the requested operation within the operation timeout window. Retrying the operation may recover this error. Execute four retries. If the error persists, it cannot be recovered.

05 *Reserved.*

3.1.4.1 Completion Code Descriptions (continued)

Code(H)	Description
06	*Hard Error:* Data error on a Read or Write command. Retry the operation. If retries fail, either the media is damaged or a drive hardware failure occurred. See your drive manual for possible causes.
07	*Interface Parity Error:* The 472 detected a parity error on the Pertec interface.
08-0D	*Reserved.*
0E	*Slave ACK Error (Non-existent memory):* The memory addressed by the 472 fails to respond within the Slave ACK timeout window. Validate the memory address or memory itself and retry the command.
0F-13	*Reserved.*
14	*Write-protect Error:* A write operation is attempted on a write-protected tape reel. Install the write enable ring and retry the Write Operation.
15	*Unimplemented Command:* The command issued is reserved or has an invalid subfunction code. Do not retry the operation.
16	*Drive Off-line:* The controller cannot perform the requested operation because the selected drive is off-line. If the tape is properly loaded and the drive on-line indicator is on, check these possible causes: bad or improperly connected *A* cable or no drive of the specified unit number is connected to the 472.
17-19	*Reserved.*
1A	*Self Test A Failure:* Either the microprocessor or its internal RAM failed the diagnostics.
1B	*Self Test B Failure:* Either the microprocessor or address counter failed the diagnostics.
1C	*Self Test C Failure:* Either the microprocessor or data buffer test failed the diagnostics.
1D	*Tape Mark Failure:* The 472 did not detect a tape mark during a Write Tape Mark operation. Retry the operation or, if enabled, the subfunction retries up to four times. If the error does not clear, erase that section of tape and retry the operation.
1E	*Tape Mark Detected:* The 472 detects a tape mark on a Read operation. It does not transfer data. System software must determine the appropriate action.

3.1.4.1 Completion Code Descriptions (continued)

Code(H)	Description
1F	*Corrected Data:* Tape drive error correction corrected a single track error during a Read command. This is a soft error. System software may log it for informational purposes only.
20–21	*Reserved.*
22	*Record Length Short:* Record length is shorter than expected. Part of the read buffer is not used. This is not necessarily an error. IOPB Bytes 10 and 11 contain the actual number of bytes read. If necessary, correct the byte count and retry the Read operation. In either case, the 472 stops in the interrecord gap.
23	*Record Length Long:* Record length is longer than expected. The 472 transfers only the requested byte count. This is not necessarily an error. IOPB Bytes 10 and 11 contain the actual number of bytes read. If necessary, correct the byte count and retry the Read operation. In either case, the 472 stops in the interrecord gap.
24–2F	*Reserved.*
30	*Reverse into BOT:* The 472 detected a Beginning of Tape (BOT) mark while moving in the reverse direction. Tape motion ceases.
31	*EOT Detected:* The 472 passes an End of Tape mark in the forward direction.
32	*ID Burst Detected:* The 472 detected a PE or GCR identification burst while reading or writing from BOT.
33	*Data Late Detected:* Indicates serious system bus contention. On a write, the 472 FIFO underflowed. On a read, the FIFO overflowed.
34	*Bad ID Field Detected:* The 472 encountered a hard error while reading the ID burst. It checks for *data late* and *record length short* errors before issuing a reverse into BOT.

3.1.5 Status Byte 3 (IOPB Byte 4)

This byte contains eight drive status flags. The flag bits are actual tape drive interface lines that the 472 updates on a command-by-command basis. If AUD is set, the 472 updates the drive status flags upon command completion.

7	6	5	4	3	2	1	0
End of Tape	Beginning of Tape	Write-protect	Rewind-ing	On-line/ Off-line	Drive Ready	Data Busy	Form. Busy

Bit	Mnemonic	Description
7	EOT	**End of Tape:** Sets when the 472 encounters an EOT mark. EOT remains set for any commands issued beyond the EOT mark. System software determines further actions.
6	BOT	**Beginning of Tape:** Sets when the 472 encounters a BOT mark. Tape is now at load point.
5	FPT	**Write-protected:** Sets if the mounted tape is write-protected.
4	REW	**Rewinding:** Sets if the selected tape drive is rewinding.
3	ONL	**On-line/Off-line:** Sets if the selected drive is on-line.
2	RDY	**Drive Ready:** Sets if the drive is ready to accept a command.
1	DBY	**Data Busy:** Sets if the selected drive encounters data.
0	FBY	**Formatter Busy:** Sets if the selected drive is executing a command. FBY remains set until the 472 completes the command.

3.1.6 Interrupt Mode (IOPB Byte 5)

7	6	5	4	3	2	1	0
Hard Error	Int. on each IOPB	GCR/ NRZ	File Mark Detected	Data Late Detected	Record Length Long	Record Length Short	Cor- rected Error

Bit	Mnemonic	Description
7	HER	**Hard Error:** The 472 sets HER when a hard tape error occurs.
6	IEI	**Interrupt on Each IOPB:** When interrupts are enabled and IEI is set, the 472 interrupts each time it completes an IOPB or sets AACK in the CSR. When interrupts are enabled and IEI is clear, the 472 only interrupts after completing the chain.
5	GCR	**GCR ID Burst or NRZ Detected:** When set, GCR indicates one of two tape densities: the tape is GCR (6250BPI) if PEID is also set, or NRZ (800 BPI) if PEID is not set.
4	FMK	**File Mark Detected:** The 472 sets FMK when it detects a file mark during a Read or File Mark Search command, or when it successfully writes a file mark.
3	DLD	**Data Late Detected:** The 472 sets DLD when it detects a *data late* condition.
2	RLL	**Record Length Long:** The 472 sets RLL when it detects a *record length long* error during a Read command.
1	RLS	**Record Length Short:** The 472 sets RLS when it detects a *record length short* error during a Read command.
0	CER	**Corrected Error:** The 472 sets CER when the tape drive indicates it corrected a media flaw.

3.1.7 Throttle (IOPB Byte 6)

The Throttle byte selects the number of DMA cycles in a DMA burst, and word or byte mode data transfers.

7	6	5	4	3	2	1	0
Transfer Mode	Reserved				Throttle Setting		

Bit	Mnemonic	Description
7	BWM	**Transfer Mode:** Selects either word or byte DMA transfers between the 472 and system memory, allowing the 472 to operate with word- and byte-oriented memory mixtures. Set BWM when reading or writing 8-bit bytes in memory. Clear BWM when reading or writing 16-bit words.
6-3		**Reserved.**
2-0	THRO	**Throttle:** Selects the maximum number of DMA cycles the 472 executes each time it becomes bus master. The throttle value determines the DMA burst length for both data and IOPB DMA transfers.

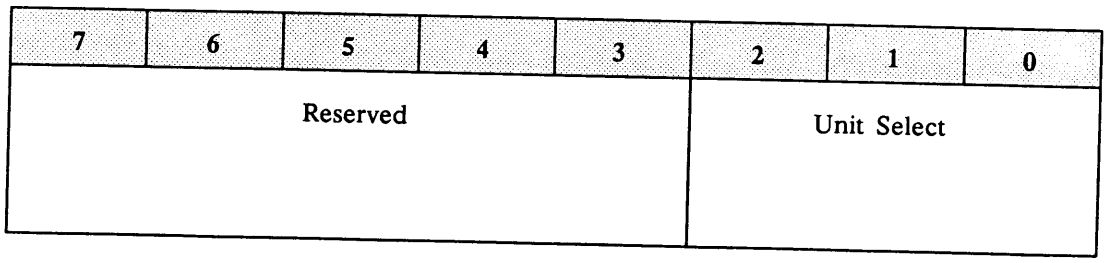
Table 3-6. Throttle Settings

Bit Value	DMA Cycles	Avg. Bus Use
0	2	1.6 µs
1	4	3.0 µs
2	8	5.8 µs
3	16	11.4 µs
4	32	22.6 µs
5,6,7	64	45.0 µs

The above bus use times were calculated using a 300 ns memory XACK response time during a Write operation. The following formula determines the average bus use for other memory XACK response times:

(XACK Response Time + 400 ns [472 overhead]) x Number of DMA Cycles + 200 ns (busack/req) = Bus Use (µsec)

3.1.8 Unit Select (IOPB Byte 7)



Bit	Mnemonic	Description
7–3		**Reserved.**
2–0	UNIT	**Unit Select:** Contains the physical unit number of one of the eight possible tape drives the IOPB can access.

3.1.9 Count (IOPB Bytes 8 and 9)

The 472 transfers tape data in bytes. The count value, a 16-bit number stored as two bytes in the IOPB, is the number of bytes for transfer. Byte 8 of the IOPB is the least significant half of the Count byte; Byte 9 is the most significant half. With a 16-bit Count byte, the 472 can execute a continuous transfer of up to 65,535-bytes with one IOPB.

System software uses the count field with the Position commands. These bytes specify the number of records or tape marks for spacing.

3.1.10 Data Address (IOPB Bytes A and B)

The data address includes two bytes in the IOPB. Byte A is the low byte; Byte B is the high byte. The data address is the starting memory address for a data transfer. The 472 adds the 16-bit data address to a shifted data relocation word to form the physical starting address of a data transfer.

3.1.11 Data Relocation Pointer (IOPB Bytes E and F)

The data relocation pointer comprises two bytes in the IOPB. Byte C is the low byte, and Byte B is the high byte. When forming a physical address, the 472 uses the Data Relocation bytes and Data Address bytes to create Multibus addresses. The 472 ignores the Data Relocation bytes if RELO is clear in the IOPB Command byte.

3.1.12 Next IOPB Address (IOPB Bytes 12 and 13)

If CHEN is set, system software must specify the next IOPB's starting address. The 472 combines Bytes E and F with the IOPB Relocation Registers to determine the Next IOPB Address. They are the links in the IOPB chain.

Byte E is the low byte, and Byte F is the high byte, of the Next IOPB Address. These two bytes comprise a 16-bit address similar to the IOPB Address Registers. The 472 adds the Next IOPB Address to the IOPB Relocation Registers to form a physical address. This physical address is 20- or 24-bits long, depending on the addressing mode jumper, and points to the next IOPB in the chain.

All IOPBs in a chain are relative to the same relocation address and must be within a 64K-byte block of memory. System software enables command-chaining by setting CHEN in the Command byte of each IOPB, except the last. Software ignores Bytes E and F if CHEN is clear.

3.1.13 Count Actually Read

During a Read command, the 472 posts the number of bytes actually read in Bytes 10 (low) and 11 (high). During a Space command, it posts the number of records actually spaced over. During a Skip Tape Mark command, the 472 posts the number of files actually skipped. These bytes help determine how many bytes the 472 transferred before returning a *record length short* status code.

> *A file is a group of one or more records terminated by a tape mark.*

4

Commands

4

Commands

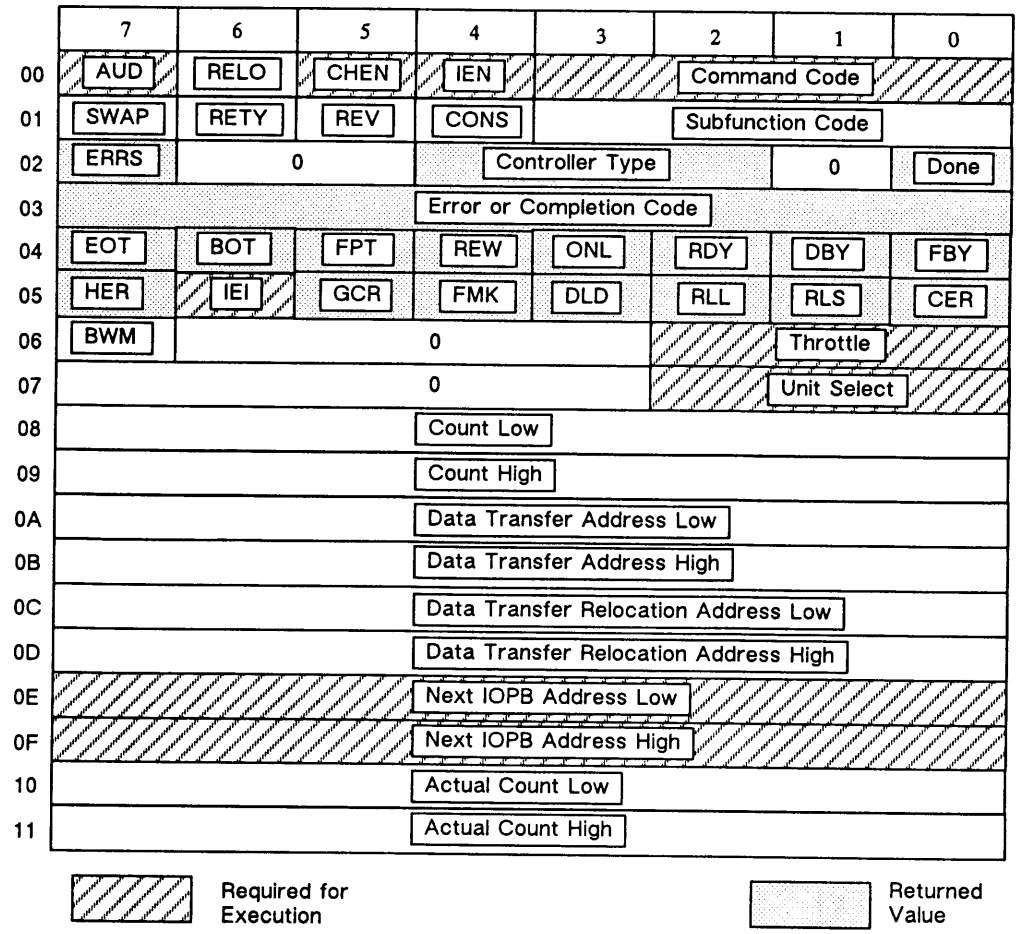
4.0 General

An IOPB diagram follows each command description. The diagrams indicate which bytes the 472 requires for command execution and which bytes it returns after execution.

The four least significant bits of the Command byte are the IOPB Command bits. These four bits enable up to sixteen possible commands (seven of which are reserved for future use). Subfunction codes allow additional flexibility in command definition.

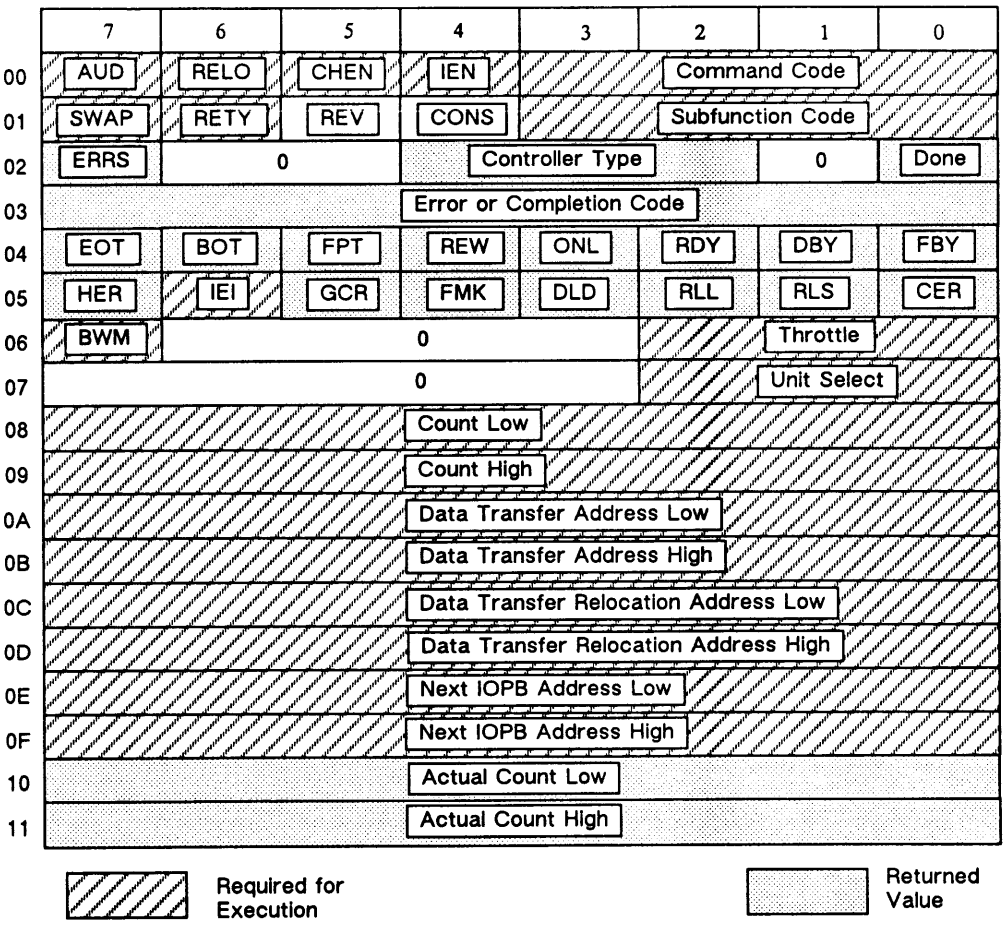
4.1 No Operation (Command Code 0)

The 472 selects a tape drive, saves DRDY (CSR bit 0), and releases the drive.



4.2 Write Data (Command Code 1)

The Write command transfers a record of data from the memory buffer, pointed to by the IOPB transfer address and relocation words, to the tape. The IOPB count field defines the record length.



4.2.1 Write Command Processing

The 472 aborts the Write command if the tape is write-protected, the drive is off-line, or system software specifies an illegal subfunction code. The tape drive and DMA start simultaneously; the 472 buffer is partially full by the time the drive needs the first byte. The 472 continues the DMA activity until it transfers the specified byte count. Tape writing continues until the buffer is empty. The 472 then waits for a *drive not busy* status, and checks the *EOT* and *read after write* status.

When the 472 reaches an EOT mark, system software should execute an End of Volume routine. The 472 does not prevent reading or writing past an EOT mark, however it sets the EOT flag and issues a hard error code which breaks an IOPB chain.

4.2.2 Write Error Handling

The 472, if enabled by the subfunction code, retries a Write operation until the error clears or it exhausts the retry count (four). The latter results in a hard error and halts IOPB processing. During each Write Retry, the 472 automatically backspaces over the block in error, erases three inches of tape, and rewrites the data block.

4.2.3 Throttle

The throttle is the maximum number of transfers allowed each time the 472 becomes bus master. The first DMA bursts on a write are at a maximum specified throttle value until the 472 fills the buffer. After data starts moving to the tape, the typical burst is less than the throttle value. The 472 decrements the byte count to zero as it transfers the required data.

4.2.4 Completing a Transfer

The 472 decrements the byte count as it transfers the specified number of bytes. After the 472 completes the transfer, it updates the IOPB status bytes and generates an interrupt, if enabled. If AUD is set, the 472 updates the IOPB, the Actual Count bytes, and the Data and Relocation Address bytes. The Data and Relocation bytes point to the last buffer location plus one (see Figure 4-1).

The 472 stops a transfer that ends in a hard error, updates Status Bytes 1 and 2, and generates an interrupt, if enabled. Any chained operations halt. The IOPB Address and Relocation Registers point to the IOPB that caused the error. The 472 updates the IOPB if AUD is set.

If the transfer ends with a *status* (soft) error, the 472 updates Status Bytes 1 and 2, and generates an interrupt, if enabled. Any chained operations continue. The 472 updates the IOPB if AUD is set. The 472 does not set ERRS or ERR.

Write a 9-byte record with a starting address of 1000H:

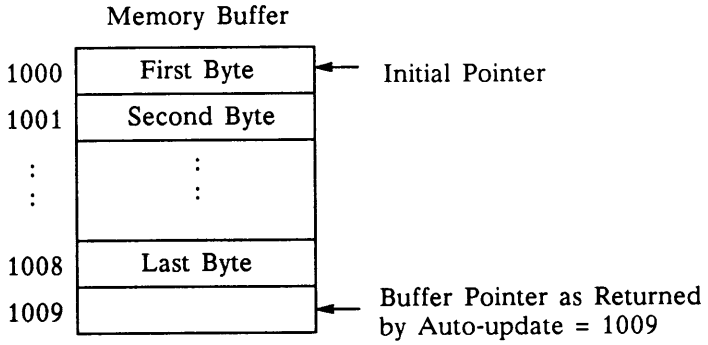


Figure 4-1. Write Buffer Pointer after Auto-update

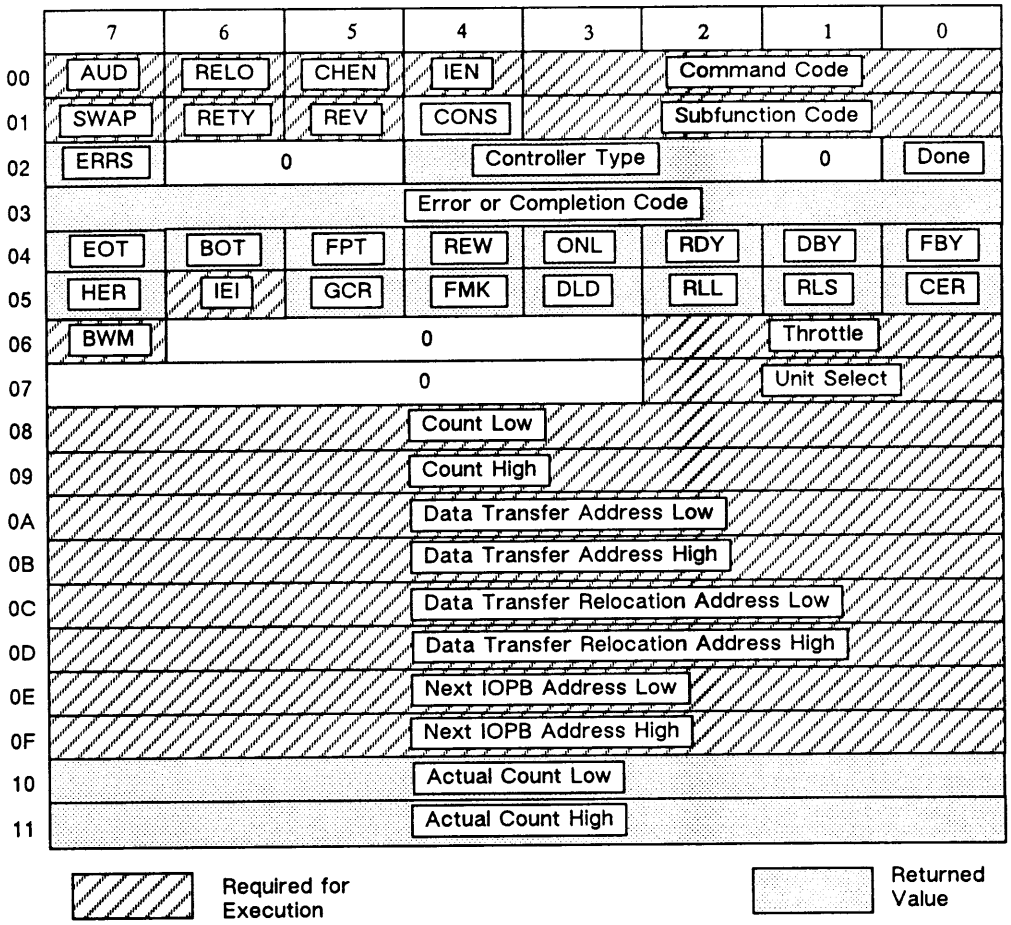
4.2.5 Write Command Subfunction Codes

Code	Description
00	*Write:* Transfers data from memory to tape according to the IOPB address and byte count. Retries are disabled, and IOPB processing halts if an error occurs.
01*	*Write Edit:* Writes a record of the same length into the record that is currently written. Writing a record no more than three times ensures proper gap spacing. The new record count must be the same size as the record you are replacing. A Read Reverse Edit must precede a Write Edit command or the 472 reports an error (see Section 4.3.4, Code 21). Issuing another command clears Write Edit.
40	*Write (With Retry):* The 472 automatically attempts up to four retries if the Write operation fails. A hard error occurs if the 472 exhausts the retry count without clearing the error. System software must clear this condition before continuing the operation.
80	*Write Swap Bytes:* The 472 swaps the most significant byte of each word with the least significant byte. This allows compatibility with machines that order their bytes in a word opposite that of the Multibus.
C0	*Write Swap Bytes (With Retry):* Swap Byte format with retries enabled.

* *Use only with microcode revisions G and up*

4.3 Read Data (Command Code 2)

The 472 transfers data from the current record on tape to the memory buffer pointed to by the IOPB Address and Relocation bytes.



4.3.1 Read Command Error Handling

The 472 aborts the Read command if the drive is off-line or system software specifies an illegal subfunction code. During a Read Previous command, the 472 transfers data starting at the top of the memory buffer. Addresses decrement rather than increment, resulting in the record being *right side up* in memory.

The drive starts and the 472 times out the *drive busy* status in case the record was written at the wrong density. Then it times out the DMA operation in case the record length is shorter than the byte count, in which case, a *record length short* status occurs, and the 472 posts the actual bytes read. If no errors occur, the 472 completes the DMA when the byte count overflows. A *record length long* status occurs if the on-board buffer is not empty.

The 472 waits for a *drive not busy* status and checks the status of *actual bytes read, corrected error, EOT* or *reverse into BOT, hard error, record length short* or *long*, and *tape mark detected*. If the 472 encounters an EOT mark, software should limit further Read commands to avoid running off the end of the tape. If the 472 encounters a Reverse Into BOT mark, software should not issue further Read Reverse commands as the tape drive will not read reverse beyond a BOT mark. The 472 does not transfer any data if it detects a tape mark. If retries are enabled, the 472 automatically retries a read until the error clears or it exhausts the retry count (four). Exhausting the retry count causes a hard error and terminates IOPB processing.

4.3.2 Throttle

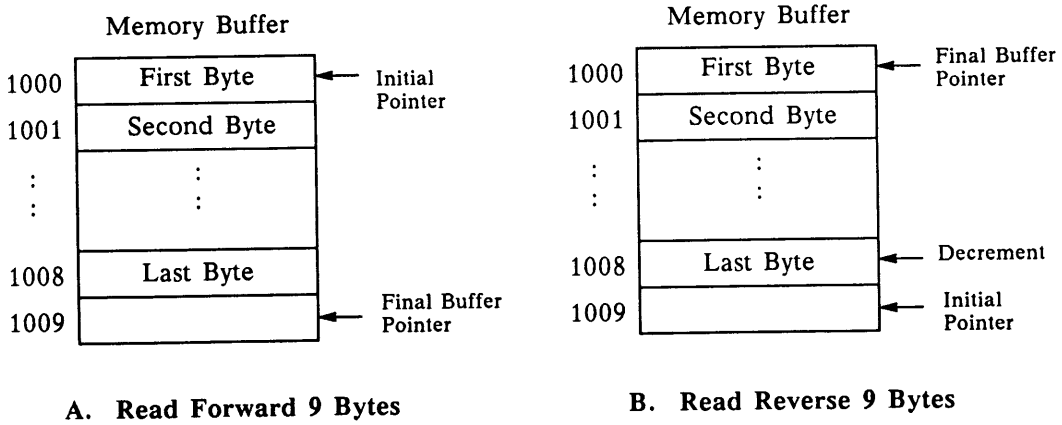
The throttle is the maximum number of transfers allowed each time the 472 becomes bus master. The first DMA bursts on a Read operation are at minimum value since the limiting factor of the DMA burst length is the number of words available from the FIFO. After the initial few DMA bursts, the typical burst length increases, possibly approaching the throttle limit. The 472 decrements the byte count to zero as it transfers the required data.

4.3.3 Completing a Transfer

When the 472 completes a transfer, it updates the IOPB status bytes and generates an interrupt, if enabled. If AUD is set, the 472 updates the IOPB along with the Actual Count and Data Relocation bytes. During a Read Forward command, the Data Relocation bytes point to the last buffer location plus one, just like the Write command (see Figure 4-2A). During a Read Reverse command, the pointer decrements by one (software transparent), and points to the last location in the buffer before the transfer begins. After the 472 completes the transfer, the Data and Relocation bytes point to the first location in the buffer (see Figure 4-2B).

The 472 stops a transfer that ends in a hard error, updates Status Bytes 1 and 2, and generates an interrupt, if enabled. Any chained operations halt. The IOPB Address and Relocation Registers point to the IOPB that caused the error. The 472 updates the IOPB if AUD is set.

If the transfer ends with a soft error, the 472 updates Status Bytes 1 and 2, and generates an interrupt, if enabled. Any chained operations continue. The 472 updates the IOPB if AUD is set. The 472 does not set ERRS or ERR.



A. Read Forward 9 Bytes

B. Read Reverse 9 Bytes

Figure 4-2. Read Buffer Pointers after Auto-update

4.3.4 Read Command Subfunction Codes

Code	Description
00	*Read Next*: Reads data in the forward direction from tape to memory as specified by the data address and byte count. Any error aborts IOPB processing. The buffer address points to the lowest address in the buffer.
20	*Read Previous:* Reads data in the reverse direction from tape to memory as specified by the data address and byte count. Any error aborts IOPB processing. The buffer address points to the highest address plus one in the buffer.
21*	*Read Reverse Edit:* Issue this command with the required record count before issuing the Write Edit command (see Section 4.2.5, Code 01). This command positions the head in front of the record being edited. Issuing any command other than a Write Edit after this command clears the Edit function.
40	*Read Next (With Retry):* Same as Read Next, but with retries enabled. A hard error occurs if the 472 exhausts the retry count. The 472 stops IOPB processing.
60	*Read Next Swap Bytes:* Swaps the position of each pair of bytes in a word as read from the tape.
A0	*Read Previous Swap Bytes:* Swaps the position of each pair of bytes as read from the tape.
C0	*Read Next Swap Bytes (With Retry):* Same as Read Next Swap Bytes, but with retries enabled.
E0	*Read Previous Swap Bytes (With Retry):* Same as Read Previous Swap Bytes, but with retries enabled.

* *Use only with microcode revisions G and up*

4.3.4 Read Command Subfunction Codes (continued)

Table 4-1 and Table 4-2 indicate the memory positions for the bytes as they are read from or written to the tape. In these examples, the data bytes in the block are numbered starting with zero. Byte 0 is always the data byte at the beginning of the block (i.e., the end of the block that is closest to the BOT mark).

Table 4-1. Byte Swap Sequence in Forward Tape Direction (Read or Write)

Swap Bytes = 0 **Buffer Address = 1001H** **Byte Count = 8**	**Swap Bytes = 1** **Buffer Address = 1001H** **Byte Count = 8**
1 0 1000 3 2 1002 5 4 1004 7 6 1006	0 1 1000 2 3 1002 4 5 1004 6 7 1006

Swap Bytes = 0 **Buffer Address = 1001H** **Byte Count = 8**	**Swap Bytes = 1** **Buffer Address = 1001H** **Byte Count = 8**
0 - 1000 2 1 1002 4 3 1004 6 5 1006 7 - 1008	- 0 1000 1 2 1002 3 4 1004 5 6 1006 7 - 1008

When reading in reverse, the first data byte on the tape is the highest numbered byte. On a Read Reverse command, the 472 stores data in the buffer from top to bottom (high address to low address). The result is the same as reading forward.

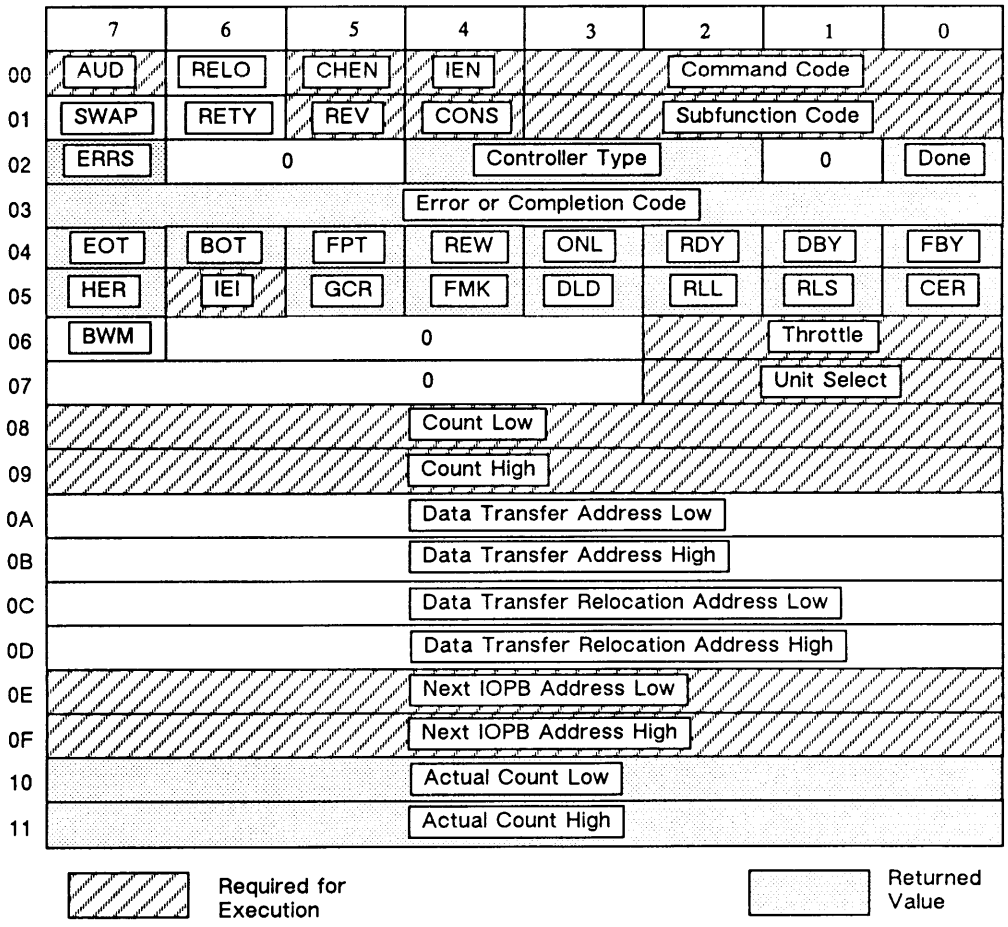
4.3.4 Read Command Subfunction Codes (continued)

Table 4–2. Byte Swap Sequence in Reverse Tape Direction (Read)

Swap Bytes = 0 Buffer Address = 1008H Byte Count = 8	Swap Bytes = 1 Buffer Address = 1008H Byte Count = 8
1 0 1000 3 2 1002 5 4 1004 7 6 1006	0 1 1000 2 3 1002 4 5 1004 6 7 106
Swap Bytes = 0 **Buffer Address = 1009H** **Byte Count = 8**	**Swap Bytes = 1** **Buffer Address = 1009H** **Byte Count = 8**
0 – 1000 2 1 1002 4 3 1004 – 7 1008	– 0 1000 1 2 1002 4 5 1004 7 – 1008
Swap Bytes = 0 **Buffer Address = 1007H** **Byte Count = 7**	**Swap Bytes = 1** **Buffer Address = 1007H** **Byte Count = 7**
1 0 1000 3 2 1002 5 4 1004 – 6 1006	0 1 1000 2 3 1002 4 5 1004 6 – 1006
Swap Bytes = 0 **Buffer Address = 1008H** **Byte Count = 7**	**Swap Bytes = 1** **Buffer Address = 1008H** **Byte Count = 7**
0 – 1000 2 1 1002 4 3 1004 6 5 1006	– 0 1000 1 2 1002 3 4 1004 5 6 1006

4.4 Position (Command Code 5)

The Position command moves tape forward and reverse, as specified by the subfunction code and byte count, without transferring data. The 472 aborts the Position command if the drive is off-line or system software specifies an illegal subfunction code.



4.4.1 Position Command Subfunction Codes

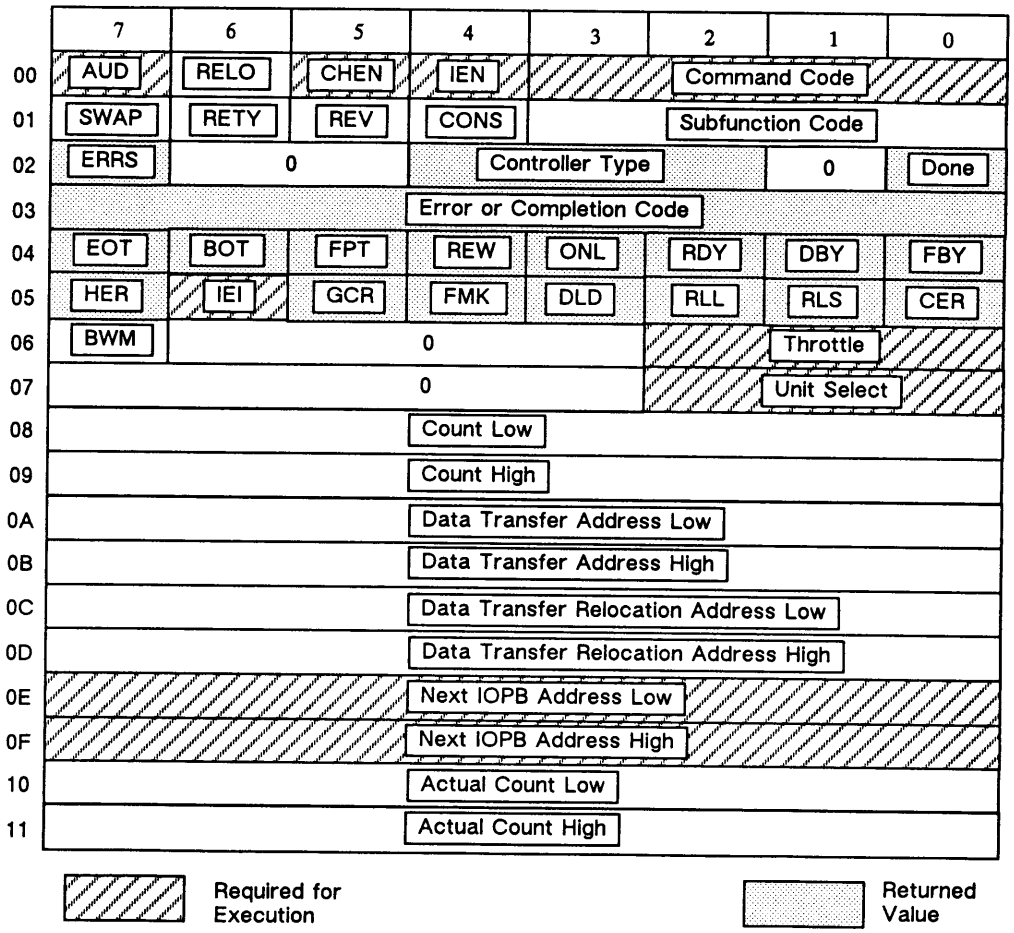
Code	Description
00	*Space Records Forward:* Spaces forward the number of records specified in the IOPB count field or until the 472 encounters a tape mark. The 472 posts a *record length short* status when it detects a tape mark and tape motion ceases. If AUD is set, the 472 posts the actual number of records spaced over in IOPB Bytes 10 and 11. If the 472 detects an EOT mark, it sets the EOT flag in Status Byte 3, posts a hard error, and halts any chained operations. It is still possible to issue unchained Space Forward commands.
20	*Space Records Reverse:* Spaces in reverse the number of records specified in the byte count field. The 472 aborts a Space Reverse command if it detects a BOT mark. If AUD is set, the 472 sets the BOT flag in Status Byte 3 and posts the actual number of records spaced over in IOPB Bytes 10 and 11.
01	*Skip Tape Marks Forward:* Skips forward the number of tape marks specified in the IOPB Count bytes. If AUD is set, the 472 posts the actual number of tape marks skipped over in IOPB Bytes 10 and 11. If the 472 detects an EOT mark, it sets the EOT flag in Status Byte 3, posts a hard error, and halts any chained operations. Tape motion continues. To prevent skipping off the end of the tape, system software should write a tape mark at the logical end of tape.
21	*Skip Tape Marks Reverse:* Skips in reverse the number of tape marks specified in the IOPB Count bytes. A Skip Reverse command stops tape motion if the 472 detects a BOT mark. If AUD is set, the 472 sets the BOT flag in Status Byte 3 and posts the actual number of tape marks skipped in IOPB Bytes 10 and 11.
02	*Rewind:* Moves the tape at high speed in the reverse direction until the 472 detects a BOT mark. Typically, the tape overshoots the BOT mark and then positions itself at low speed exactly at the BOT mark. System software determines when the 472 completes the rewind by issuing a Get Status command. Subsequent commands to the same drive automatically wait for the rewind motion to cease.

4.4.1 Position Command Subfunction Codes (continued)

Code	Description
03	*Unload:* Moves tape at high speed in the reverse direction until the 472 detects a BOT mark. The tape continues moving at low speed until it loses tension, then the drive goes off-line. System software determines when the drive finishes rewinding and goes off-line by issuing a Get Status command. You must load a new tape and place the drive on-line before it will accept any further Position commands.
11	*Search for Consecutive Tape Marks Forward:* Skips forward searching for the number of consecutive tape marks specified in the count field. Code 11 is especially useful for positioning the tape at the logical end of tape. Typically, two consecutive tape marks indicate the logical end of tape.
31	*Search for Consecutive Tape Marks Reverse:* Same as Search Forward, except the 472 performs the operation in the reverse direction.

4.5 Drive Reset (Command Code 6)

A Drive Reset command selects and resets a drive, and aborts any operation in progress. Use Drive Reset only if you suspect a runaway condition. Always consider the tape position unknown after a drive reset.

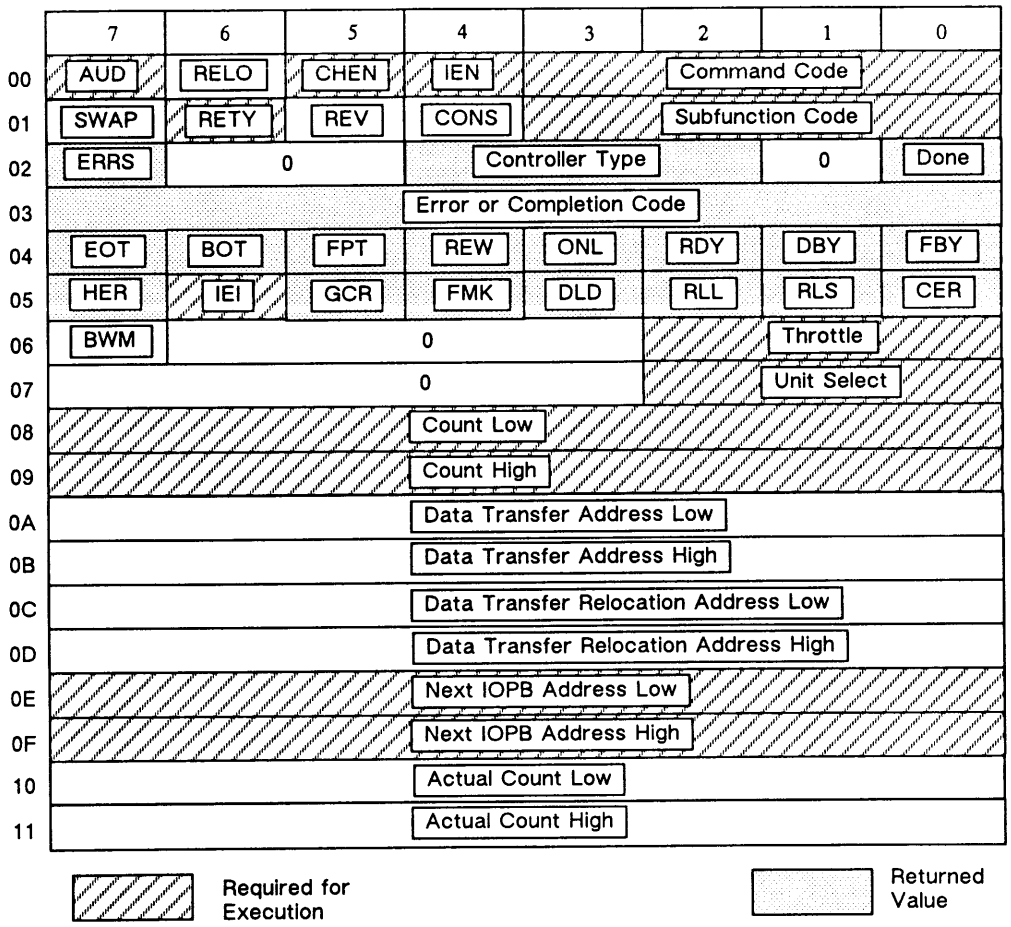


4.5.1 Drive Reset Subfunction Code

Code	Description
01	*Load On-line:* Brings the drive on-line and to load point (BOT). This is useful in remote installations if a power loss occurs. Only certain drive manufacturers support this feature.

4.6 Write Tape Mark/Erase (Command Code 7)

The Write Tape Mark/Erase command erases bad spots on the tape or writes tape marks. Tape marks logically group records; typically, software writes several consecutive tape marks at the logical end of tape to prevent spacing or skipping off the end of the tape.



4.6 Write Tape Mark/Erase (continued)

The 472 aborts the Write Tape Mark/Erase command if the tape is write-protected, the drive is off-line, or system software specifies in illegal subfunction code. If the 472 detects an EOT mark, it posts an EOT flag, but does not prevent writing past the EOT mark. If it does not detect a tape mark during a Write Tape Mark command, the 472 returns a *tape mark failure* status. The 472 retries the Write Tape Mark command until it exhausts the retry count, if so enabled by the subfunction, or detects a tape mark. Exhausting the retry count causes a hard error and halts IOPB processing.

During each Write Tape Mark Retry, the 472 automatically backspaces over the block in error, erases three inches of tape, and rewrites the tape mark.

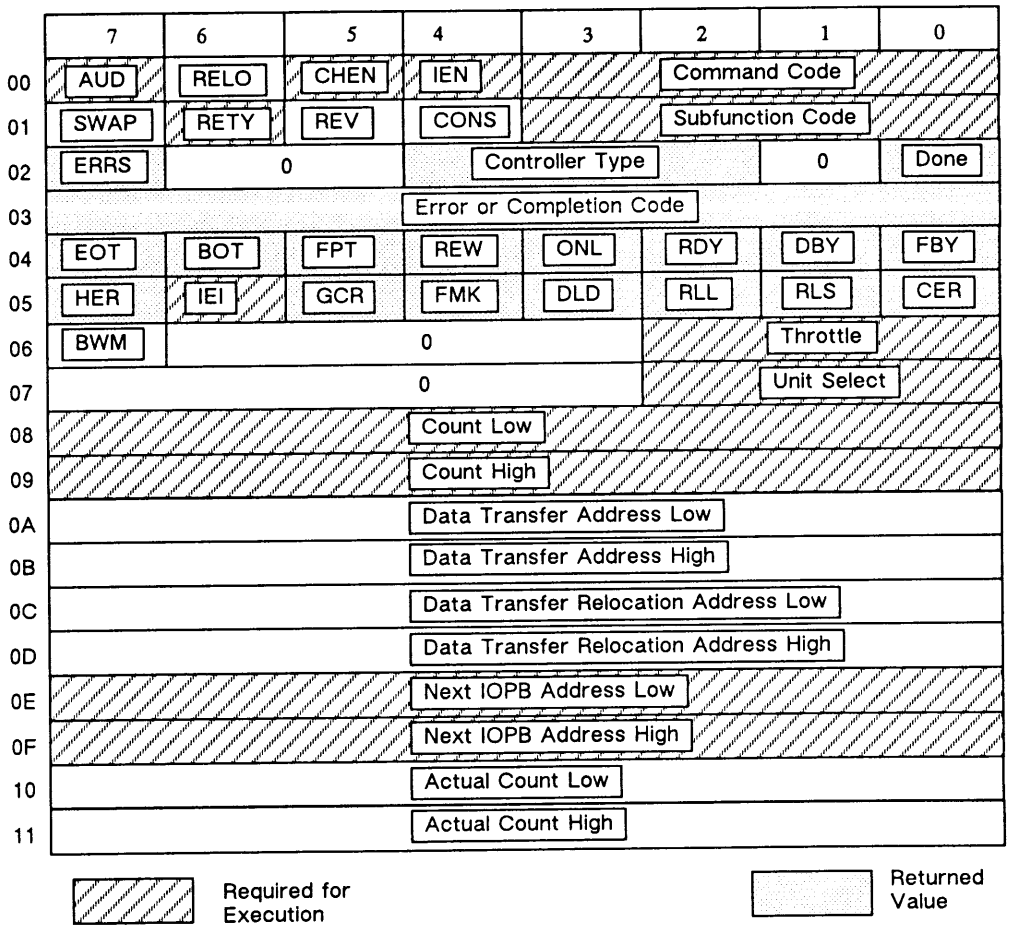
4.6.1 Write Tape Mark/Erase Subfunction Codes

Code	Description
00	*Write Tape Mark:* Writes a tape mark. Tape marks logically group records on tape. System Software should write at least on tape mark at the logical EOT.
01	*Erase:* Use this function when Write With Retries fails to write the desired data. Failure to write successfully usually results from media degradation and can be overcome by writing a long gap (IBG) over the media defect. This function erases approximately three inches of tape per command.
40	*Write Tape Mark (With Retry):* Same as Write Tape Mark, but with retries enabled. The 472 posts a hard error and aborts IOPB processing if it exhausts the retry count before recovering the operation.
02*	*Security Erase:* Use this function to erase the remainder of the tape after writing the last record; this assures that no other data records or partial records remain on the tape. Security Erase also initializes new tapes by erasing them from BOT to EOT. It accelerates tape to operating speed and, after the appropriate prerecord delay, activates the erase head. Security Erase erases tape from its present position to a point 6 feet (1.82 meters) beyond EOT. You must issue a Rewind command. *Do not issue any further commands until the tape is at BOT.*
03*	*Variable Erase:* Set the count field (Bytes 8 and 9) to the length of the erase time (1 count = 1/16 sec). The maximum count is FFFFH or 65,535 x 1/16 sec. = 255 sec. The actual length of the erase depends on the drive.

* *Use only with microcode revisions G and up*

4.7 Get Status (Command Code 9)

The Get Status command selects a drive and checks for *on-line, BOT, EOT, write-protect* and *rewind* status. The 472 posts the results of the Get Status command in Status Byte 3 if AUD is set. This command can also read back sense information from the drive. The Sense bytes are useful in determining the cause of drive malfunctions, and are for maintenance purposes only. Sense bytes are currently limited to certain manufacturer's drives. Consult your drive manual for more information.



4.7 Get Status (Command Code 9) (continued)

The 472 posts the selected drive status in Status Byte 2, and in Status Byte 3 if AUD is set. System software polls Status Byte 3 and determines if a drive is ready for a particular command (see Section 3.1.5). Some PE and GCR streamers support the Get Status command in reading back sense information from the drive. System software must set up the correct byte count in Bytes 8 and 9 (see subfunction code or drive manufacturer's manual for number of Sense bytes) and buffer address (IOPB Bytes A and B).

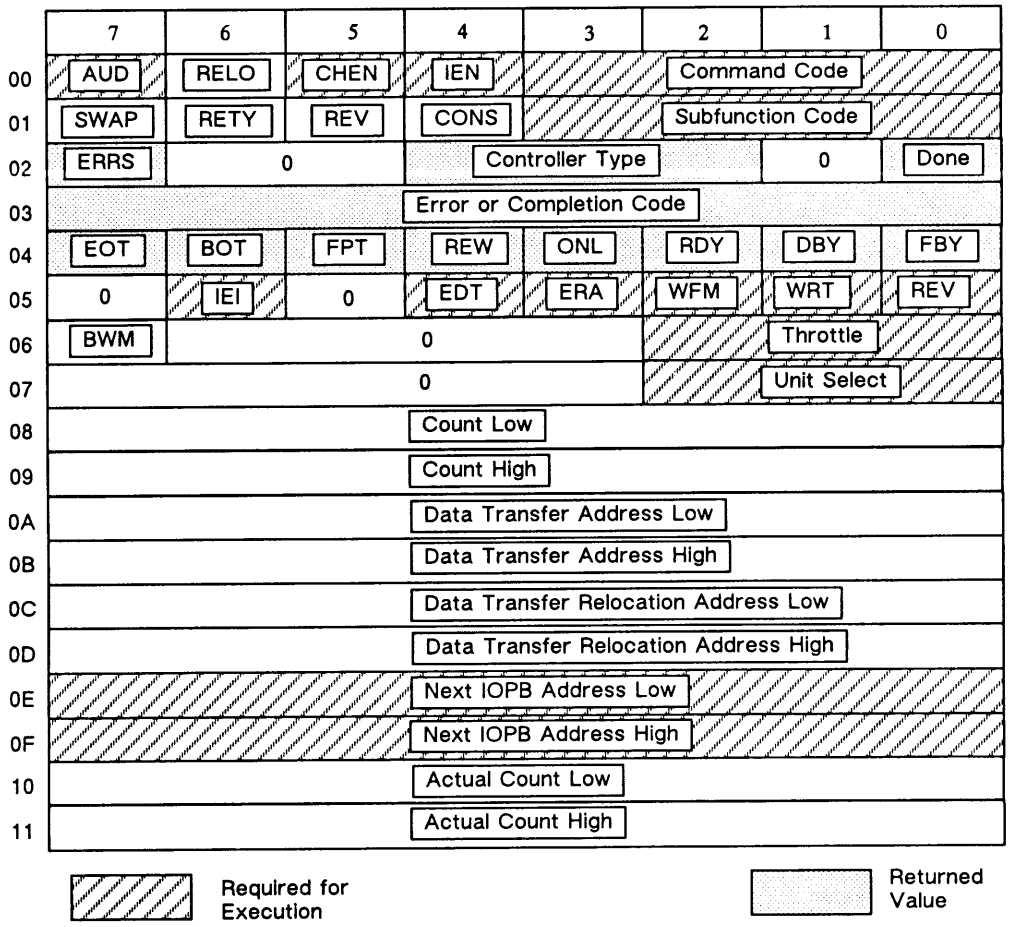
4.7.1 Get Status Subfunction Codes

Code	Description
00	*Drive Status Bytes:* Returns the current drive status in Status Byte 2, and Status Byte 3 if AUD is set.
01	*Read Sense Bytes:* Returns drive status information to a memory buffer. System software must issue the correct byte count and buffer address. Consult the drive manufacturer's manual for the number and definition of bytes.
02	*Read Extended Sense Bytes:* Returns drive status information to a memory buffer; useful in determining drive malfunctions. System software must issue the correct byte count and the data address. For example, the CDC 92185 has 27 bytes of extended sense information. Consult the drive manufacturer's manual for definition of bytes.

4.8 Set Parameters (Command Code B)

Set Parameters changes the selected drive's density, speed, or interrecord gap length. Note that the 472 uses the standard IOPB, but redefines bits in Byte 05 for this command.

The 472 aborts a density change command if the tape is not at the BOT mark (*illegal command* status). This ensures that each tape is written at a single density. The density change commands apply to both PE 1600/3200 BPI and PE 1600/GCR 6250 BPI tape drives. Use a density change command only with drives that support dual density. Some single density drives hard error if you attempt a density change.



4.8 Set Parameters (Command Code B) (continued)

The 472 accepts a speed change command anywhere on the tape. A Set High Speed command puts the tape drive into streaming mode. In this mode, consecutive reads or writes move the tape continuously at high speed as long as they meet the drive's command reinstruct time. The drive must reposition if the reinstruct time is not met.

A Set Low Speed command puts the tape drive into start/stop mode (12.5 or 25 IPS). Stop/start is more efficient over the length of a tape than streaming mode since the drive repositions on every record.

The Set Speed commands select density on non-streaming drives. Set Low Speed mode selects NRZI (800 BPI) and Set High Speed mode selects PE (1600 BPI). Configure the drive for remote control of density selection.

4.8.1 Set Parameters Subfunction Codes

Some Kennedy tape drives use the Pertec interface line J1-36 when switching density modes. When executing a Density Select, use microcode revision G or higher and substitute Long Gap Select/Disable with High/Low Density Select (subfunctions 04 [High] and 05 [Low]).

Code	Description
00	*Set Low Density Mode:* Use with dual density streaming tape drives. Sets the drive formatter into PE (1600 BPI) mode.
01	*Set High Density Mode:* Use with dual density streaming tape drives. Sets the drive formatter into 3200 BPI or 6250 BPI mode.
02	*Set Low Speed Mode:* Use with streaming tape drives; selects the drive speed (12.5 or 25 IPS, depending on manufacturer). Selects NRZI (800 BPI) on certain start/stop tape drives.
03	*Set High Speed Mode:* Use with streaming tape drives. Sets the drive into constant streaming mode (typically 75 or 100 IPS). Selects PE (1600 BPI) on certain start/stop tape drives.

4.8.1 Set Parameters Subfunction Codes (continued)

Code	Description
04*	*Long Gap Mode Select:* Places the selected unit number in long gap mode. The actual length of the long gap depends on the drive model and manufacturer.
05*	*Long Gap Mode Disable:* Clears LGAP and switches the unit into normal, or short, gap mode. *The 472 does not return a status indicating long gap mode is selected; certain drive manufacturers allow reading the Sense bytes, which indicate the last gap size selected.*
06*	*Set 800 BPI:* Currently supported by the Pertec FS2000 tape drive.
07*	*Set 3200 BPI:* Replicates the Set High Density Mode command.
08*	*Set 6250 BPI:* Supported by the Cipher 990 and Pertec FS2000.
09*	*Set 25 IPS:* Supported by Pertec FS2000.
0A*	*Set 100 IPS:* Supported by Pertec FS2000.
0B*	*Megatape Track Select:* The Count Low and Count High bytes select the track number. Each track is treated as a separate tape with its own BOT and EOT markers. If a Rewind command is issued, the drive selects Track 0 and automatically rewinds to BOT. Use Space and Skip commands to position the heads on individual tracks.
0C-0E	*Reserved.*
0F*	*Command Pass Through:* The *Reverse*, *Write*, *Write File Mark*, *Erase*, and *Edit* bits in Byte 5 allow system software to issue commands directly to the tape drive. The 472 sets the interface bits as requested, strobes IGO, and then waits for *formatter done*. Only issue parameter change commands with this feature (i.e., density, speed, etc.). Code 0F does not support commands that require a count (i.e., Read, Write). Your tape drive manual specifies the use of these lines.

* *Use only with microcode revisions G and up*

Table 4-3 shows the subfunction codes each drive manufacturer supports for selecting density, speed, and long gap. If your drive does not support these codes, then use the Command Pass Through command.

4.8.1 Set Parameters Subfunction Codes (continued)

Table 4-3. Drive Use of Command Pass Through Codes

Drive	800/NRZ	1600/PE	3200/PE	6250/GCR	HISPD	LOWSPD	LGAP
Anritsu 2500	N	00	01	N	02	03	N
Cipher 880	N	D	N	N	N	N	N
891	N	00	01	N	N	N	N
990	N	00	01	08	N	N	N
CDC 92181	N	D	N	N	02	03	04/05
92182	N	D	N	N	02	03	04/05
92185	N	00	N	01	02	03	04/05
Fujitsu 2442	N	00	N	01	02	03	04/05
Kennedy 9400	S	S	N	S	N	N	N
9600	OF+16	00	N	N	?	?	?
Pertec FS1000	N	D	01	N	02	03	X
FS2000	06	D	01	08	09	0A	X
STC 2920	N	00	N	01	02	03	04/05
Telex Shamrock	X	00	N	01	02	03	04/05
Thorn 9800	?	04	05	N	02	03	N
9900	OF+6	04	05	N	02	03	X

Key:
- N - Not supported by drive manufacturer.
- X - Not supported by Xylogics 472.
- D - Drive defaults to this density at BOT.
- S - Code depends on drive switch settings.
- / - Enable/Disable
- + - Requires subfunction code plus Command Pass Through field.
- * - Not documented.
- ? - Not known.

4.8.2 Tape Command Lines

Xylogics uses certain bits when implementing formatter commands via the tape command lines. Table 4-4 lists each formatter command that Xylogics supports, and the bits that each command enables.

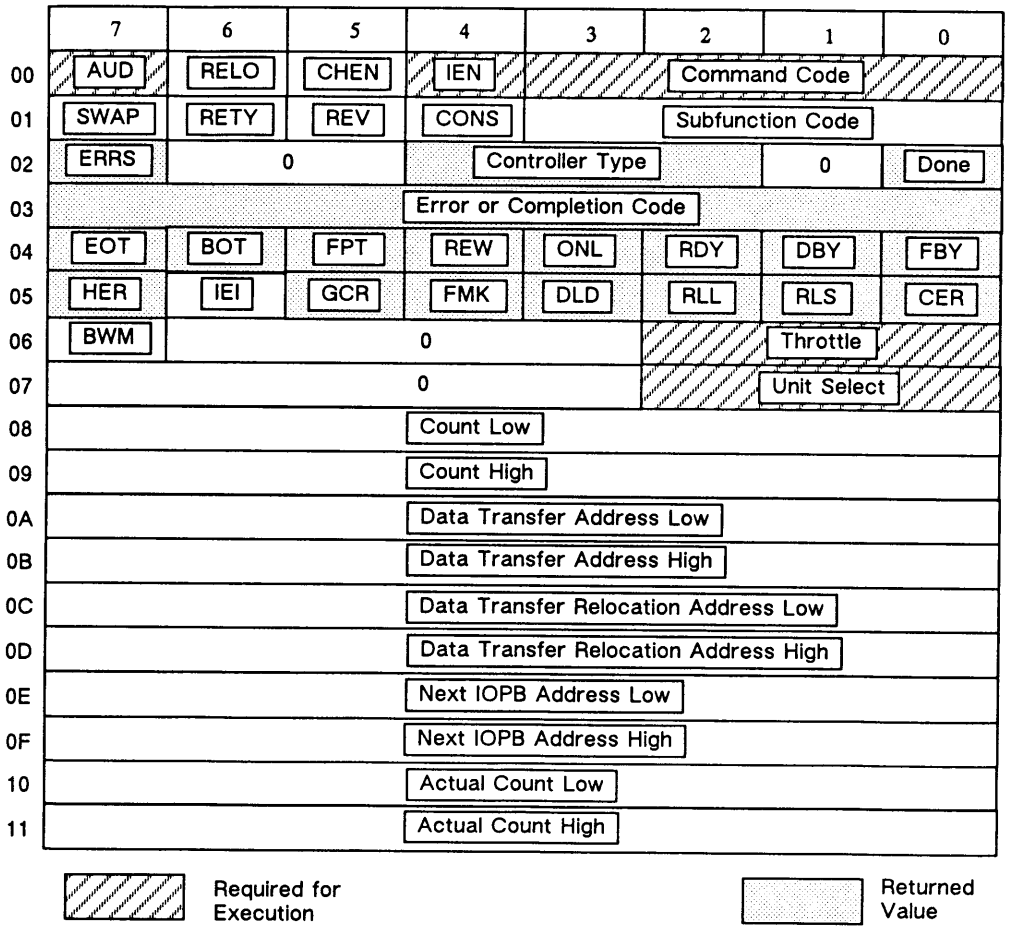
Table 4-4. Tape Command Lines

Formatter Command	Reverse	Write	Write File Mark	Edit	Erase
Read Forward	0	0	0	0	0
Read Reverse	1	0	0	0	0
Write	0	1	0	0	0
Write File Mark	0	1	1	0	0
Erase	0	1	1	0	1
Space Forward	0	0	0	0	1
Space Reverse	1	0	0	0	1
Read Sense	1	0	0	1	1
Read Extended Sense	1	1	0	1	1
Select Low Density Mode	0	0	1	1	1
Select High Density Mode	1	0	1	1	1
Security Erase	0	1	1	1	1
Variable Length Erase	0	1	0	0	1
Formatter Track Select	1	1	0	1	0
Select 800 BPI	1	1	0	0	1
Select 3200 BPI	1	0	1	1	1
Select 6250 BPI	1	1	0	0	0
Select 25 IPS	1	1	1	0	1
Select 100 IPS	1	1	1	1	0

4.9 Self Test (Command Code C)

The Self Test command starts the same self test the 472 runs automatically on power-up. The 472 posts a *success* status if it successfully completes the test. Otherwise, it posts the appropriate *error* status. Do not use the Self Test command if CHEN is set.

The Self Test LED goes on at the start of the test and off after the 472 successfully completes the test. The LED stays on if the 472 detects a failure. The 472 posts the self test error codes in Status Byte 2. The following subsections describe the self tests.



4.9.1 Processor RAM Test

This test verifies that the microprocessor and its internal RAM are functioning properly. The 472 posts the error code 1AH in Status Byte 2 if it fails this test.

4.9.2 Address Counter Test

This test verifies that the address counter and the microprocessor are functioning properly. The 472 posts the error code 1BH if it fails this test.

4.9.3 Buffer Test

This test checks the on-board FIFO buffer for faulty memory. The 472 posts the error code 1CH if it fails this test.

4.9.4 Commands 3, 4, 8, A, D, E, and F

These commands are reserved.

5

Programming the Controller

5

Programming the 472

5.0 General

This section describes methods for programming the 472. Assume that interrupts are enabled. Ignore the references to interrupting if they are not enabled in your situation.

5.1 Individual IOPB Processing

- **Set Up the IOPB**

  Allocate an 18-byte long segment of memory for the IOPB. Set the various bytes in this IOPB as required to perform a function (see Section 3).

- **Point the 472 to the IOPB**

  Write the IOPB address into the 472 IOPB Address Registers.

- **Set the Go/Busy bit**

  Writing an 80H (GBSY) to the CSR starts the operation. The host processor either polls the CSR for *Done* or waits for the interrupt. When polling, wait for a valid status code (see Section 2.3.6).

  The 472 processes the IOPB after GBSY sets in the CSR. It uses the Address and Address Relocation Registers to address Multibus memory and read the IOPB. The 472 executes the function, updates the IOPB Status bytes, clears GBSY, and interrupts.

- **Check for Errors**

  System software reads both the CSR and Status Byte 1 and determines if the 472 successfully completed the command. Then it tests the CSR and determines if DERR is set; this may indicate that Status Byte 2 was not updated. If DERR is clear, software should check ERR. If ERR is clear, the 472 successfully completed the command.

5.1 Individual IOPB Processing (continued)

- **Check for Errors (continued)**

  The 472 posts a completion code for a command in Status Byte 2. A code of zero indicates successful completion; any other value indicates a status change or an error occurred. Section 3.1.4 details the completion codes; Section 5.4 explains the error recovery procedures.

5.2 IOPB Chain Processing

The chain starts with the IOPB pointed to by the IOPB Address Registers and follows the address pointers in each IOPB to the next IOPB. The 472 completes all IOPBs or stops processing the chain when a hard error occurs.

5.2.1 The Chain

Each IOPB has a field that points to the next IOPB in the chain. The 472 does not look at the chain pointer unless CHEN is set. The 472 uses the IOPB Relocation Registers to relocate the Next IOPB Address bytes to point to the next IOPB. All IOPBs in a chain must be located within the 64K-byte memory block starting at the base address in the IOPB Relocation Registers.

5.2.2 Executing the IOPB Chain

Set CHEN and write the address of the next IOPB into Bytes 12 and 13 of the last IOPB in the current chain. The new IOPB is now the last one in the chain. The next address field of the last IOPB in the chain must equal zero, and CHEN must be clear. Once the 472 starts processing an IOPB chain, it is considered busy. System software must not add or remove IOPBs from the chain without executing the attention protocol. The 472 does not update the IOPB in memory, excluding Status Bytes 1 and 2, unless software sets AUD.

5.2.3 Completing an IOPB

The 472 updates the IOPB Status bytes as it completes each IOPB. If *Interrupt on Each IOPB* (IEI) is set, the 472 interrupts as it completes each IOPB. Software acknowledges an interrupt by writing a one to the *Interrupt Pending* (IPND) bit in the CSR. Do not clear IPND, ERR, or DERR with a *Controller Reset* since the 472 terminates IOPB processing and may modify certain operating parameters.

GBSY remains set until the 472 completes the chain or a hard error occurs. If the 472 completes processing its current IOPB while there is an active *Attention Request*, clearing AREQ sets GBSY again.

5.2.4 Modifying the Chain During Execution

The 472's attention protocol uses two bits in the CSR: *Attention Request* (AREQ) and *Attention Acknowledge* (AACK). System software sets AREQ to notify the 472 that it wishes to add or remove IOPBs from the chain. When the 472 recognizes this request, it sets AACK. If IEI is set, the 472 interrupts after setting AACK.

System software may now remove those IOPBs marked *Done*, and/or may add new IOPBs to the chain. You may modify CHEN and the Next IOPB Address, but do not touch previously chained IOPBs that are not marked *Done*.

5.2.5 Restarting a Modified Chain

When system software finishes adding or removing IOPBs, it clears AREQ, and the 472 clears AACK. Then the 472 continues processing the chain. Whenever reading the CSR, system software must check for a valid status code.

5.2.6 Chain Interrupts

The 472 provides a single interrupt at the end of each IOPB: an interrupt occurs after the 472 completes on IOPB, completes several chained IOPBs, or sets AACK. System software must determine why the interrupt occurred.

5.2.7 Completing a Chain

The chain is complete when the 472 processes all the IOPBs. The 472 terminates the chain with an error if one IOPB has a hard error, and generates an interrupt, if enabled. It may not process later IOPBs.

5.3 Performance Considerations

This section suggests how to get the best possible performance from the 472 for your particular application. It discusses the various tradeoffs, their advantages and disadvantages.

5.3.1 Streaming

The 472 has several features that facilitate streaming: high speed DMA sequencing, DMA throttle, and a 2K-byte, or optional 8K-byte, FIFO buffer. A slow disk controller or inadequate software can make streaming difficult. The disk controller should have a DMA transfer rate sufficiently higher than the disk transfer rate to allow time for the 472 to DMA from memory to tape. Streaming can be achieved as long as the combined transfer rates of the disk and tape add up to less than the DMA capacity of the system, and no other devices need the bus during the transfer. An SMD disk transfers at 1.2 MBS and the GCR streamer transfers at 0.5 MBS. The combined rate of 1.7 MBS is far below the throughput capability of the Multibus using 300 ns memory and Xylogics controllers (e.g., 3.0 MBS). You must recompute this figure for slower disk controllers or slower memory. Where the transfer rate capabilities of the two controllers differ, compute a composite transfer rate by multiplying the percentage of bus time needed by each controller.

The transfer rate considerations define a theoretical ability to stream the tape. To actually stream the tape, software must meet the tape drive's command reinstruct time (typically 3.5 milliseconds [ms]). Sending the next tape command within this 3.5 ms window is easy, providing the controller writes valid data. Typically, system software sets up a data buffer in memory to hold disk data waiting to be written to the tape. Thus the stream of data is uninterrupted, even if the disk must access a new track or skip a bad sector. Helpful programming options include sequential sector reads, sector interleaving, adjusting the size of the tape record, and adjusting the size and number of the data buffers. Parity and ECC errors cause the tape to reposition. Generally, streaming operations limit file processing. For best results, execute a *data compare* between the disk and tape in start/stop mode.

5.3.2 Data Buffering

The key to efficient system integration of streaming tape drives is matching the dissimilar and variable transfer rates of the tape and disk. One way software solves this problem is by using appropriate multiple buffer schemes. The maximum disk latency that a streaming mode operation supports determines the size of the buffer. One buffering scheme has several buffers arranged as a circular chain. Memory buffers, along with on-board FIFO buffers, provide a steady data flow to the streaming tape avoiding frequent stopping for new data.

5.3.3 Throttle

From the 472's viewpoint, the throttle value should be as high as possible, so it never has to skip revolutions. You may have a real-time application that must access the bus periodically. In these applications:

- Determine the maximum time the 472 can be bus master (less time than another unit can be without the bus).
- Determine your memory's response time, add 400 ns, and divide that figure into the allowable 472 bus master time. The result is the maximum throttle value. Reduce this number for smaller throttles to allow for getting on and off the bus.
- Pick the figure closest to the 472 throttle value without going over the actual amount.

The advantage of a high throttle value is:

- Maximum bus throughput with minimum bus overhead.

The disadvantages of a high throttle value are:

- Tendency to *hog* the bus; time critical devices fail.
- Other DMA units may not get enough bus time.

5.3.4 Word or Byte Mode

Word mode is definitely more efficient than byte mode on the bus. Since it takes the same length of time to transfer a word in word mode as it does a byte in byte mode, using word mode effectively doubles the 472's throughput.

5.3.4 Word or Byte Mode (continued)

The advantages of word mode are:

- Increased throughput with less bus utilization.

  Helps DMA keep up with the disk.

The disadvantage of word mode is:

- Works only on word-oriented memory.

5.3.5 Transfers on Address Boundaries

The 472 reacts differently to transfers on various address boundaries. Word mode transfers on odd addresses must compensate for the odd address. The 472's internal architecture dictates how it handles transfers across page boundaries.

5.3.5.1 Word Mode on Odd Address Boundary

If a word mode transfer begins on an odd address, the 472 compensates by mixing word and byte mode. It performs the first transfer in byte mode, the remaining transfers in word mode, and completes the transfer with the final byte in byte mode. The disadvantage is the on-board microprocessor must get involved with the transfers more often when starting a word mode transfer on an odd byte. This added involvement requires more time during a transfer and reduces the margin for keeping up with the tape. The extra time involved by the microprocessor does not result in any extra bus time.

5.3.5.2 Transfers to Page Addresses

Each time the 472 crosses a 256-byte address boundary, the on-board microprocessor updates the upper address bits and restarts the DMA sequencer. Aligning transfers on page boundaries minimizes microprocessor involvement.

5.3.6 Command-chaining

Command-chaining results in several performance advantages: The 472 may automatically initiate overlap seeking, dramatically increasing performance throughput in multidrive systems. System software does not have to respond as rapidly at the end of a command; the 472 continues the next command without any operating system intervention. The 472 interrupts at the end of each IOPB and notifies the system that the IOPB is complete.

6

Installing and Testing the Controller

6

Installing and Testing the 472

6.0 General

This section describes how to unpack, configure, install, and test your 472 controller.

6.1 Unpacking and Inspection

6.1.1 Inspect the Shipping Carton and the Controller

Inspect the carton for possible shipping damage. If there is damage, do not unpack the unit. Notify Xylogics and the freight carrier immediately. If no damage is visible, carefully unpack the 472. Save the carton and other packing materials for future use.

6.1.2 Contents

The 472 is a single printed circuit board. Optional items include a manual and software on a magtape or floppy disk.

If any items are missing or damaged, please contact Xylogics at one of the following telephone numbers:

United States (Burlington, MA): (617) 272-8140;
United Kingdom (Milton Keynes): 44-908-222112.

6.1.3 Handling Precautions

Observing proper handling precautions minimizes the risk of damaging the 472 with electrostatic discharge. When transporting the 472, use an antistatic bag, antistatic bin, or the original shipping carton and packing material. Personnel handling the controller should observe proper grounding methods including, but not limited to, wrist bands, heel straps, and antistatic mats.

6.2 Configuring the 472

You can configure the 472 with several jumper options. The following paragraphs describe these options.

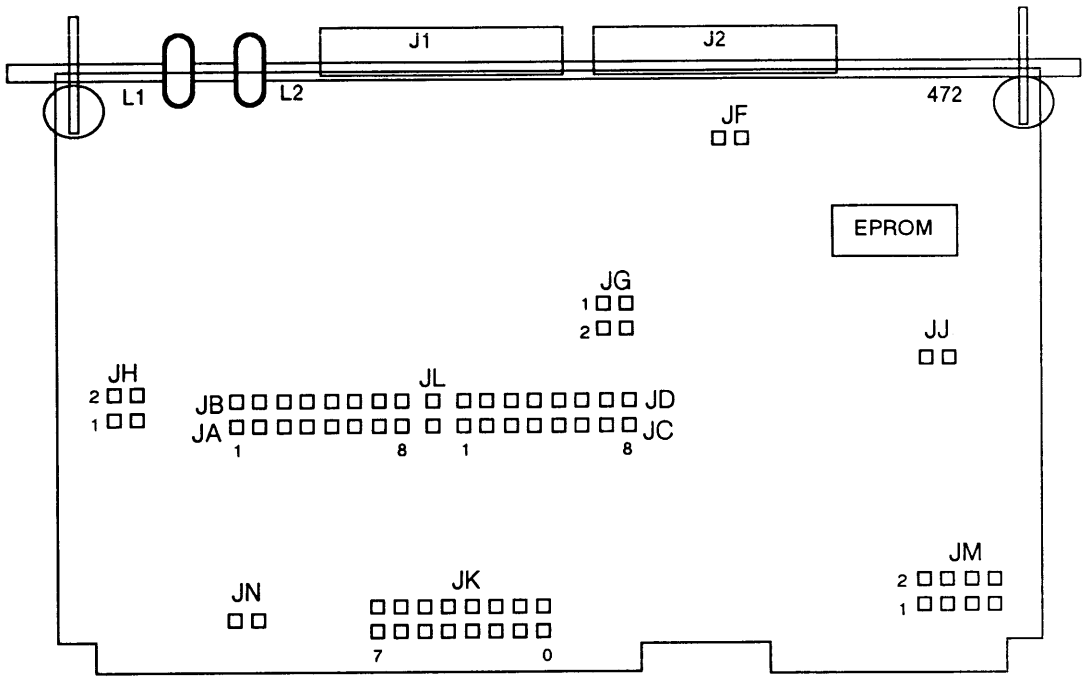


Figure 6-1. Jumper Locations

6.2.1 Base Address Selection

There are two parts to selecting the base address. First, select a response to 8- or 16-bit register addresses. Jumper JL controls this option. Jumper JL *in* indicates 8-bit addressing; JL *out* indicates 16-bit addressing. Factory configuration: jumper JL *out*.

Second, jumpers JA, JB, JC, and JD control the base address. If you select 8-bit addressing, JC/JD 1–8 are the only valid jumpers; ignore jumpers JA/JB 1–8 . Figure 6–3 shows how to set the jumpers for commonly used base addresses. You cannot jumper the upper eight Multibus address bits (ADR10–ADR17). Factory configuration: EE60.

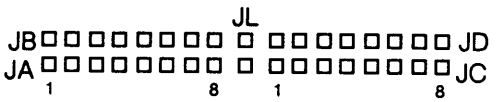


Figure 6–2. Base Address Jumpers

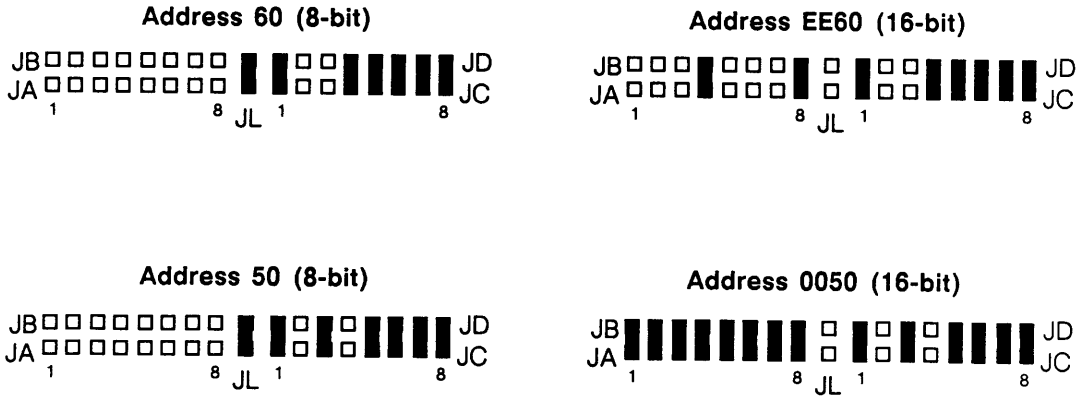


Figure 6–3. Jumpering the Base Address

6.2.2 20/24-Bit Address Relocation

The 472 functions in backplanes of 16-, 20-, and 24-bit addresses. Jumper JF selects the 20- or 24-bit addressing mode. Software selects the 16-bit addressing mode. System software determines the status of the jumper by reading ADMD in the CSR. When set, the board is jumpered for 24-bit addressing (and jumper JM must have all pins *in*). Both 20- and 24-bit addressing modes support 16-bit addressing. Factory configuration: jumper JF *out*.

Mode	Jumper JF	ADMD
16/20	Remove	0
16/24	Install	1

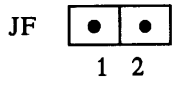


Figure 6-4. Jumper Block JF

6.2.3 24-Bit Extended Addressing

For systems not using 24-bit addressing, jumper JM disconnects those signals from the 472's P2 connector. Factory configuration: jumper JM 1-4 *in*.

Address Bit	Jumper JM
ADR17/	1
ADR16/	2
ADR15/	3
ADR14/	4

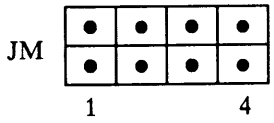


Figure 6-5. Jumper Block JM

6.2.4 Interrupt Request Levels

You can choose any one of eight interrupt request levels. To select an interrupt level, connect jumper JK as per Table 6-1. Factory setting: jumper JK 5 *in*.

Table 6-1. Interrupt Request Levels

Interrupt Level	Jumper JK
INT0/	0
INT1/	1
INT2/	2
INT3/	3
INT4/	4
INT5/	5
INT6/	6
INT7/	7

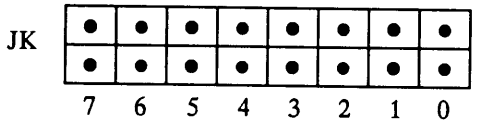


Figure 6-6. Jumper Block JK

6.2.5 Disable Bus Priority Out

If you are using the 472 in parallel DMA arbitration (see Section 6.3.2.2), isolate the *Bus Priority Out (BPRO/)* signal from the Multibus by removing jumper JN. Factory configuration: jumper JN *in*.

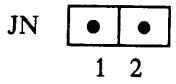


Figure 6-7. Jumper Block JN

6.2.6 DMA Sequencer Clock

The source for the DMA sequencer clock comes from either an on-board 10-MHz crystal or *BCLK/* on the Multibus backplane. Typically, the sequencer clock connects to the on-board 10-MHz crystal.

If *BCLK/* runs at 10-MHz, the sequencer clock synchronizes to it, maximizing the 472's throughput. If *BCLK/* is slower or faster than 10-MHz, connect the sequencer clock to the 10-MHz crystal. Factory Configuration: jumper JH 2 *in*.

Clock	Jumper JH
10-MHz	2
BCLK/	1

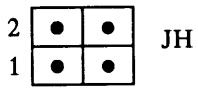


Figure 6-8. Jumper Block JH

6.2.7 Buffer Size

The size of the on-board FIFO buffer determines the jumper settings. Factory configuration: by order.

Buffer Size	Jumper JG	Jumper JJ
2 KB (Standard)	1 - In 2 - Out	Out
8 KB (Optional)	1 - Out 2 - In	In

Figure 6-9. Jumper Blocks JG and JJ

6.3 Preparing the Computer System

The backplane of your system must provide a Multibus slot for the 472. The slot must be capable of handling a bus master, and the power source must handle the power consumption of the entire system, including the 472.

6.3.1 Card Cage Slot

The card cage must have a slot available for the 472. Placement of the 472 in the DMA priority chain may be critical; consider this when choosing a slot.

6.3.2 DMA Bus Arbitration

The 472 uses either serial or parallel DMA arbitration. Serial arbitration is much easier to implement, but has restrictions on the number of bus masters it can arbitrate. Parallel bus arbitration is harder to implement, but is more versatile and can handle more bus masters. Priority schemes are system-dependent and have no impact on the 472's performance.

6.3.2.1 Serial DMA Priority

To implement serial priority, connect the *BPRO/* and *BPRN/* lines in a serial fashion (see Figure 6–10). The first slot has the highest priority, and must have its *BPRN/* line grounded. The next slot has the next highest priority. A unit must have its *BPRN/* line asserted to become bus master. If a unit is not currently a bus master, it passes the state of *BPRN/* to *BPRO/*. If the unit is bus master, it deasserts its *BPRO/* so the following units cannot assert their *BPRN/* lines and become bus master.

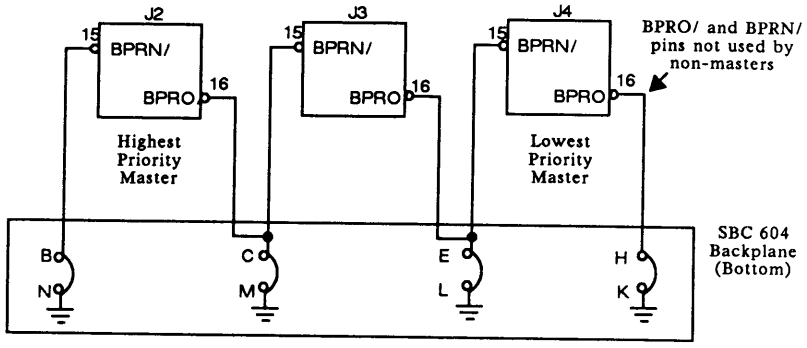


Figure 6–10. Serial DMA Priority

6.3.2.2 Parallel DMA Priority

Parallel and serial priority use the same connections to each board. To implement parallel priority, connect these signals to an external circuit similar to that of Figure 6-11. This circuit implements bus arbitration. Disconnect the *BPRO/* lines between each board and connect the *BREQ/* lines to the external priority resolver circuit (see Section 6.2.5).

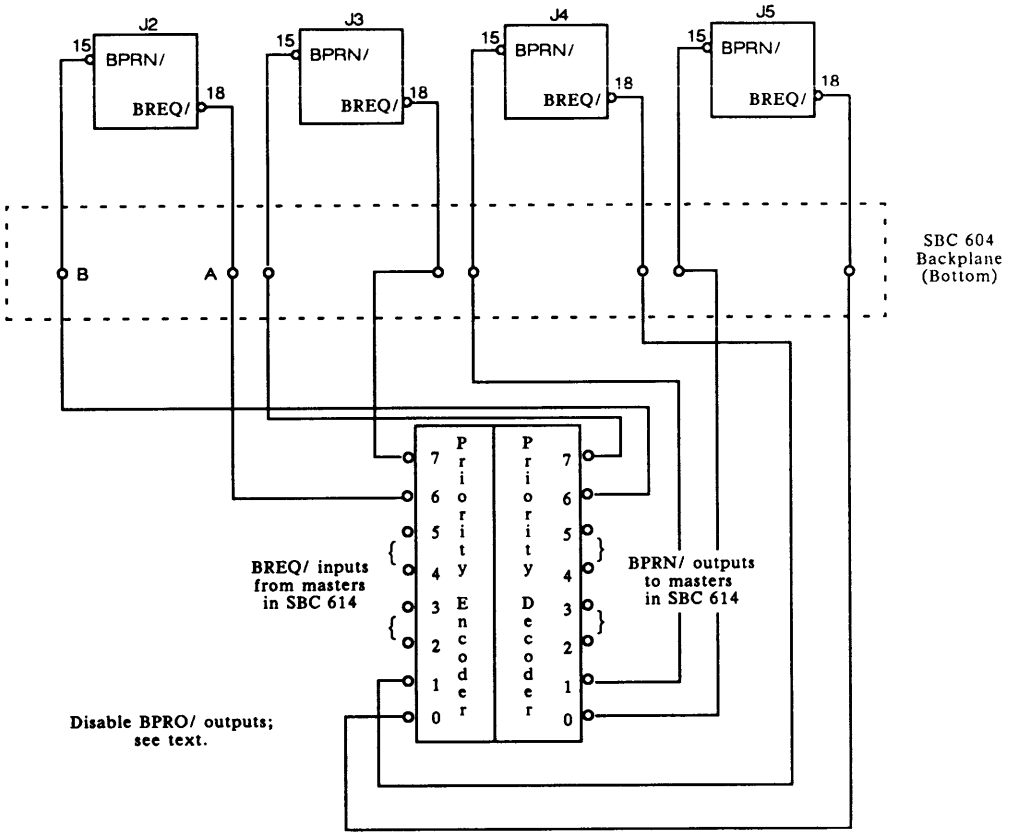


Figure 6-11. Parallel DMA Priority

6.3.3 Power Considerations

The 472 affects the power consumption of the entire computer system. Make sure the power supplies can handle the entire power load. Readjust the voltages *after* plugging in the 472. A power supply that is just adequate may cause intermittent and unusual problems from noise as it occasionally goes into overcurrent protection. The 472 draws 5.0 amps at 5.0 VDC.

6.4 Tape Drive Preparation

Unpack the tape drive and configure it for use with the 472. This entails setting up several parameters such as formatter address, unit select, remote density select, etc. Your drive manual provides specific configuration information.

6.5 Install and Cable the 472

Place the 472 into the computer card cage; make sure it is firmly seated. Do not dislodge any socketed ICs. Situate the tape drive and connect it to the appropriate power source.

6.5.1 Connect the Cables

A cable set consists of two identical 50-pin flat ribbon cables that conform to the Pertec-format interface standard; these cables are typically 15-feet long (see Section 7 for the tape drive interface pinout). Check the tape drive manufacturer's manual for any interface adapters you may need. Observe the *pin 1* markings on the cable connector for proper orientation on both the drive and the 472. Using *pull tabs* on the cables greatly reduces connector damage.

6.5.2 Mechanical Restraint

Mechanically restraining the cables at each end prevents accidental disconnection.

6.5.3 Tape Drive Grounds

Install a ground braid wire between the ground terminal on the tape drive(s) and the computer system ground.

6.6 Power-up and Self Test

The 472 initiates a self test upon power-up. The LED L1 lights up, and then goes off. If it remains on, the board is not functioning properly. Contact Xylogics for further assistance.

6.6.1 Register Verification

Check the 472 registers. On power-up the registers clear to all zeros. Verify this by reading them. If system software is unable to access the registers, check the 472 base address jumpers and/or your system's I/O Register requirements. Try writing some value to the registers and read them back to verify the data is correct. If software still cannot access the registers, check your system's address relocation requirements (this is a common problem).

After determining that the 472 registers are functional, execute a NOP command. The NOP tests the 472 hardware and firmware.

6.6.2 Tape Drive Diagnostics

Many tape drive manufacturers offer on-line and off-line diagnostics. Consult your tape drive manual for further information.

6.6.3 Drive On-line

Load a tape reel, press the load switch, and wait for the on-line indicator to light. Read the 472 Reset Register to reset the controller. Select Drive 0 and test the *drive on-line* status. Next, read the CSR; it contains the results of the *drive on-line* check. The CSR should read 01H or 09H if jumpered for 24-bit addressing.

If bit 0 is clear, check the drive cable connections and try again. If you are still unable to get the proper status, check the tape drive with the off-line diagnostics or tester.

6.7 Cabling Multiple Drives

Connect the first drive in the chain directly to the 472; connect additional drives together, starting with the initial drive (for example, the 472 connects to Drive 0; Drive 0 connects to Drive 1; Drive 1 connects to Drive 2, etc.) Be careful; do not reverse the cables. Terminate the cabling at the last drive in the chain. You can daisy-chain up to eight tape drives in this manner (see Figure 6-12 and Figure 6-13).

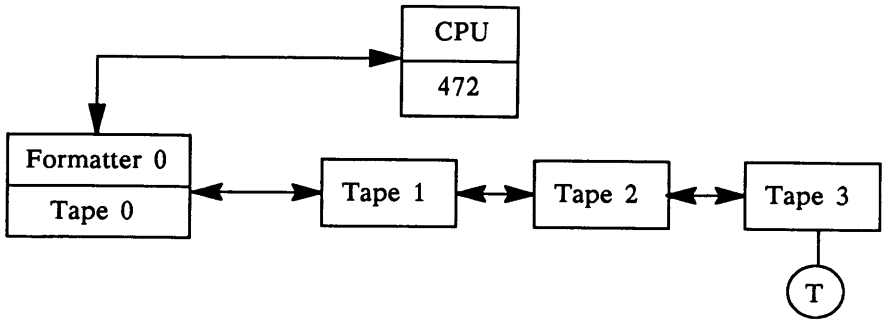


Figure 6-12. Tape Drive Daisy-chaining (Single Formatter)

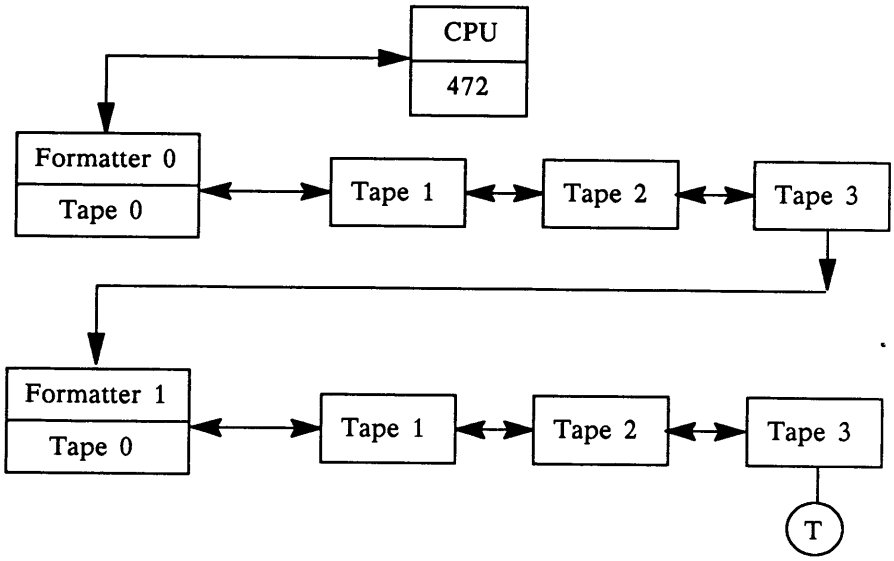


Figure 6-13. Tape Drive Daisy-chaining (Two Formatters)

6.7.1 Formatter/Unit Select

The 472 requires one formatter board for every four connecting tape drives. The formatter is part of the drive electronics. Systems using more than four drives require an additional formatter board. Each formatter has an address that is selected via switches. The tape drives must have unique unit numbers that are selected via switches in the drive. Consult the drive manufacturer's manual for the location and proper setting of the switches.

6.8 Cache Tape Drives

6.8.1 Data Late Errors

The 472 assumes that the DMA is faster than the typical tape ramp-up time; it starts the tape first (IGO), and then kicks off the DMA. Cache tape drives can request data from the 472 immediately following the IGO pulse. If there is no ramp delay time, and system memory is too slow, the tape beats the DMA to the FIFO buffer and a *data late dectected* error occurs. Configuring the drive for a ramp delay eliminates this problem; the procedure differs from one manufacturer to the next.

6.8.2 Operation Timeout

When nearing EOT, certain cache tape drives automatically reduce the size of the cache buffer; this keeps the tape from running past the physical end of the tape. The 472 may time out the operation if it does not see the *Data Busy* signal in time. If this timeout occurs, software should retry the operation.

6.8.3 Setting the Transfer Rate

Most cache drives allow you to select the drive's transfer rate (KBS). The range is from 100 KBS to 1 MBS, depending on the manufacturer. The 472 can handle any of these speeds since it DMAs at 3 MBS. However, if system memory is too slow, the FIFO buffer can overrun on tape reads or empty on a tape write causing a *data late* error. If this occurs, lower the tape transfer rate to match the overall system speed.

6.8.4 Missing a Record?

If the 472 does not report an error, but you are missing one of the data records that you wrote, the following scenario probably occurred, assuming retries are enabled: the 472 writes data into the cache memory; before the cache writes to the tape the drive reports a hard error due to a hardware fault. The 472 assumes the hard error is a media problem and retries the write but, in fact, the record was never really written to tape. Then, during a typical retry, the 472 issues a Space Reverse and rewrites the record; in this case it is the previous good record. Disabling retries and querying the tape drive as to the type of hard error that occurred via Read Sense commands eliminates the problem.

6.9 Long Gap

The 472 supports the Long Gap feature via the tape interface pin J1-36. Be careful; certain manufacturers use this line for selecting density. Several manufacturers (Kennedy 9600, Pertec FS2000, and Thorn 9000 series) also select Long Gap via Pin J1-44; the 472 does not support this pin selection.

6.10 Non-latched End of Tape

Several of the Cipher 100 series and the Cipher 920 do not latch EOT when passing the marker, but provide an EOT pulse. Since the standard 472 has no latching mechanism, it misses the fact that EOT was detected (it expects EOT to remain active once passed). A special hardware version of the 472 is available if you are using any of these drives.

6.11 Hard Error

Typically, the tape interface line *Hard Error* (J2-12) indicates a media flaw that you cannot correct. In such a case, it is reasonable to expect the 472 or software to issue a retry and resume operation. Many late model tape drives use this line to indicate anything from a hard tape error to a drive hardware error. Many of these drives provide front panel displays that indicate the error type. Illegal command codes or command sequences commonly cause hard errors. In such cases it is possible to recover from the error, but the recovery procedure is drive-dependent. It may not be possible to recover via software from a hard error caused by a drive failure, in which case, you must manually reset the drive (possibly power-down). Depending on the failure, you still may not recover from the error upon power-up.

6.12 Record Length Error During Read Reverse

The CDC 92185 cache drive has a feature that allows executing a Read Reverse command as a Space Reverse command. A *record length* error occurs if you attempt a Read Reverse when this feature is enabled. Turn off this feature if you are using a Read Reverse or a XYCAT test board. See the CDC Manual regarding CE Test 84 for more information.

> *Most tape drive manufacturers do not support the Read Reverse command when the drive is in GCR (6250 BPI) density. If attempted, the 472 reports a record length error or the drive may report a hard error.*

7

Interface Signals

7

Interface Signals

7.0 General

This section provides useful interface information for installing and maintaining your 472 streaming tape controller.

7.1 Multibus Interface Signals

Mnemonic	Conn.	Pin	Used By 472	Description
ADR0/	P1	57	Y	
ADR1/	P1	58	Y	
ADR2/	P1	55	Y	
ADR3/	P1	56	Y	
ADR4/	P1	53	Y	
ADR5/	P1	54	Y	
ADR6/	P1	51	Y	
ADR7/	P1	52	Y	
ADR8/	Pi	49	Y	
ADR9/	P1	50	Y	
ADRA/	P1	47	Y	
ADRB/	P1	48	Y	
ADRC/	P1	45	Y	
ADRD/	P1	46	Y	Address Bus
ADRE/	P1	43	Y	
ADRF/	P1	44	Y	
ADR10/	P1	28	Y	
ADR11/	P1	30	Y	
ADR12/	P1	32	Y	
ADR13/	P1	34	Y	
ADR14/	P2	57	P	
ADR15/	P2	58	P	
ADR16/	P2	55	P	
ADR17/	P2	56	P	

Y = Yes; N = No; P = Possibly (Jumper or optionally available)

7.1 Multibus Interface Signals (continued)

Mnemonic	Conn.	Pin	Used By 472	Description
DAT0/	P1	73	Y	
DAT1/	P1	74	Y	
DAT2/	P1	71	Y	
DAT3/	P1	72	Y	
DAT4/	P1	69	Y	
DAT5/	P1	70	Y	
DAT6/	P1	67	Y	
DAT7/	P1	68	Y	
DAT8/	P1	65	Y	Data Bus
DAT9/	P1	66	Y	
DATA/	P1	63	Y	
DATB/	P1	64	Y	
DATC/	P1	61	Y	
DATD/	P1	62	Y	
DATE/	P1	59	Y	
DATF/	P1	60	Y	
Strobe				
IORC/	P1	21	Y	I/O Read Command
IOWC/	P1	22	Y	I/O Write Command
MRDC/	P1	19	Y	Memory Read Command
MWTC/	P1	20	Y	Memory Write Command
XACK/	P1	23	Y	Transfer Acknowledge
DMA				
BPRN/	P1	15	Y	Bus Priority In
BPRO/	P1	16	Y	Bus Priority Out
BREQ/	P1	18	P	Bus Request
BUSY/	P1	17	Y	Bus Busy
CBRQ/	P1	29	P	Common Bus Request

Y = Yes; N = No; P = Possibly (Jumper or optionally available)

7.1 Multibus Interface Signals (continued)

Mnemonic	Conn.	Pin	Used By 472	Description
Interrupts				
INT0/	P1	41	P	
INT1/	P1	42	P	
INT2/	P1	39	P	
INT3/	P1	40	P	
INT4/	P1	37	P	Interrupt Request Levels
INT5/	P1	38	P	
INT6/	P1	35	P	
INT7	P1	36	P	
INTA/	P1	33	N	Interrupt Acknowledge
Miscellaneous Control				
BHEN/	P1	27	Y	Byte High Enable
BD RESET/	P2	36	N	Board Reset
HALT/	P2	28	N	Bus Master Wait State
INH1/	P1	24	N	Inhibit 1; Disable RAM
INIT/	P1	14	Y	Initialize
Miscellaneous				
ACLO/	P2	18	N	AC Low
ALE/	P2	32	N	Bus Master ALE
AUX RESET/	P2	38	N	Reset Switch Reserved
LOCK/	P1	25	N	Inhibit 2; Disable PROM or ROM
MPRO/	P2	20	N	Memory-protect
PAR1/	P2	27	N	Parity 1
PAR2/	P2	29	N	Parity 2
WAIT/	P2	30	N	Bus Master Wait State

Y = Yes; N = No; P = Possibly (Jumper or optionally available)

7.1 Multibus Interface Signals (continued)

Mnemonic	Conn.	Pin	Used By 472	Description
Clocks				
BCLK/	P1	13	Y	Bus Clock
CCLK/	P1	31	N	Constant Clock
PLC/	P2	31	N	Power Line Clock
Power				
12VB	P2	11,12	N	+12 VDC Battery
5VB	P2	3	N	+5 VDC Battery
GVB	P2	4	N	Return
-5VB	P2	7,8	N	-5 VDC Battery
-12VB	P2	15,16	N	-12 VDC Battery
+5V	P1	3,4,5,6,81,82,82,84	Y	+5 VDC
+12V	P1	7,8	N	+12 VDC
+15	P2	23,24	N	+15 VDC
-5V	P1	9,10	N	-5 VDC Supply
-12V	P1	79,80	N	-12 VDC
-15V	P2	25,26	N	-15 VDC
EEVPP	P2	6	N	E2 PROM Power
GND	P1	1,2,11,12,75,76,85,86	Y	Signal GND
GND	P2	1,2,21,22	N	Signal GND

Y = Yes; N = No; P = Possibly (Jumper or optionally available)

7.2 Pertec-formatted Interface Signals

Signal mnemonics differ from one manufacturer to the next, but the function of the signal remains the same. Several of the signals listed are used only with GCR streamers. Consult the manufacturer's interface specification for further explanation.

Xylogics uses 50-pin male connectors manufactured by 3M (part number 3596-5002).

Name	Cable	Signal Pin	Ground Pin	Description	Used by 472
FBY	J1	2	1	Formatter Busy	Y
LWD	J1	4	3	Last Word	Y
WD4	J1	6	5	Write Data 4	Y
GO	J1	8	7	Initiate Command	Y
WD0	J1	10	9	Write Data 0 (MSB)	Y
WD1	J1	12	11	Write Data 1	Y
Spare	J1	14	13	Reserved	N
LOL	J1	16	15	Load On-line	Y
REV	J1	18	17	Reverse/Forward	Y
REW	J1	20	19	Rewind	Y
WDP	J1	22	21	Write Data Parity	Y
WD7	J1	24	23	Write Data 7	Y
WD3	J1	26	25	Write Data 3	Y
WD6	J1	28	27	Write Data 6	Y
WD2	J1	30	29	Write Data 2	Y
WD5	J1	32	31	Write Data 5	Y
WRT	J1	34	33	Write/Read	Y
LGAP	J1	36	35	Long Gap	Y
EDIT	J1	38	37	Edit	Y
ERASE	J1	40	39	Erase	Y
WFM	J1	42	41	Write File Mark	Y
THR	J1	44	43	High Clip	N
TAD0	J1	46	45	Transport Address	Y
RD2	J1	48	47	Read Data 2	Y
RD3	J1	50	49	Read Data 3	Y
RDP	J2	1	5	Read Data P	Y
RD0	J2	2	5	Read Data 0 (MSB)	Y
RD1	J2	3	5	Read Data 1	Y
LDP	J2	4	5	Load Point	Y
RD4	J2	6	5	Read Data 4	Y

7.2 Pertec-formatted Interface Signals (continued)

Name	Cable	Signal Pin	Ground Pin	Description	Used by 472
RD7	J2	8	7	Read Data 7	Y
RD6	J2	10	9	Read Data 6	Y
HER	J2	12	11	Hard Error	Y
FMK	J2	14	13	File Mark	Y
ID	J2	16	15	PE ID Burst	Y
FEN	J2	18	17	Formatter Enable	Y
RD5	J2	20	19	Read Data 5	Y
EOT	J2	22	21	End of Tape	Y
OFL	J2	24	23	Off-line/Unload	Y
GCR	J2	26	25	Density Status	N
RDY	J2	28	27	Ready	Y
RWD	J2	30	29	Rewinding	Y
FPT	J2	32	31	File-protect	Y
RSTR	J2	34	33	Read Strobe	Y
WSTR	J2	36	35	Write Strobe	Y
DBY	J2	38	37	Data Busy	Y
HSPD	J2	40	39	Speed	N
CER	J2	42	41	Corrected Error	Y
ONL	J2	44	43	On-line	Y
TAD1	J2	46	45	Transport Address 1	Y
FAD	J2	48	47	Formatter Address	Y
HISP	J2	50	49	High Speed Select	Y

Index

Index

Index

Index

R

S

T

U

W

X

Manual Evaluation Form

Please help us produce quality documentation by filling out this form. No envelope is needed, just fold this page, stamp, and return. Thank you.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy	☐	☐	☐	☐
Completeness	☐	☐	☐	☐
Clarity	☐	☐	☐	☐
Organization	☐	☐	☐	☐
Figures (usefulness)	☐	☐	☐	☐
Tables (usefulness)	☐	☐	☐	☐
Index	☐	☐	☐	☐
Page layout	☐	☐	☐	☐

What I like best about this manual is _____

_____

_____

_____

What I like least about this manual is _____

_____

_____

_____

I found the following errors in this manual:

Page	**Description**
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions for improving this manual:

_____

_____

_____

Name/Title: _____

Company: _____

Address: _____

_____ Phone: _____

- fold here -

Xylogics, Inc.
53 Third Avenue
Burlington, MA 01803

Att: Technical Support/Documentation

- fold here -

Xylogics, Inc.
53 Third Avenue
Burlington, MA 01803